
LOAD
REVISTA PARA USUARIOS DE
MSX
AÑO 1 Nº 4 A 2.30 REP. ARGENTINA
BASE DE DATOS PARA UN GRAN MODEM
SOFTWARE PARA
LA OFICINA
LA ESCUELA
EL HOGAR
EL ORIGINAL ROBOT MSX
CINTAS Y GRABADORES
INSTRUCCIONES DE LA DISKETERA TALENT DPF-550
ML-ROBO


¡YA!
CLUB DE USUARIOS
Talent MSX
Cabildo 2027 - 1º "A" (1428) Cap. Fed.
Esmeralda 320 - 3º (1343) Cap. Fed.
Tucumán 2044 - 1º (1050) Cap. Fed.
diálogo - D.P.



AÑO 1 Nº 4



Director General
Ernesto del Castillo

Director Editorial
Cristian Pusso

Director Periodístico
Fernando Flores

Director Financiero
Javier Campos Malbrán

Arte y Diagramación
Fernando Amengual
Tamara Migelson

Coordinador
Ariel Testori

Redacción
Eduardo Mombello
Andrea Sabin Paz

Departamento de Avisos
Oscar Devoto

Departamento de Publicidad
Guillermo González Aldalur

Departamento fotográfico
Víctor Grubicy

Load Revista para usuarios de MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5º Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Registro Nacional de la Propiedad Intelectual: E.T. M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de Propiedad Intelectual.

Precio de este ejemplar: A 2,30.
Impresión: Calcotam. Fotocromo tapa: Columbia. Fotocomposición: Van Waveren.
Los ejemplares atrasados se venderán al precio del último número en circulación.

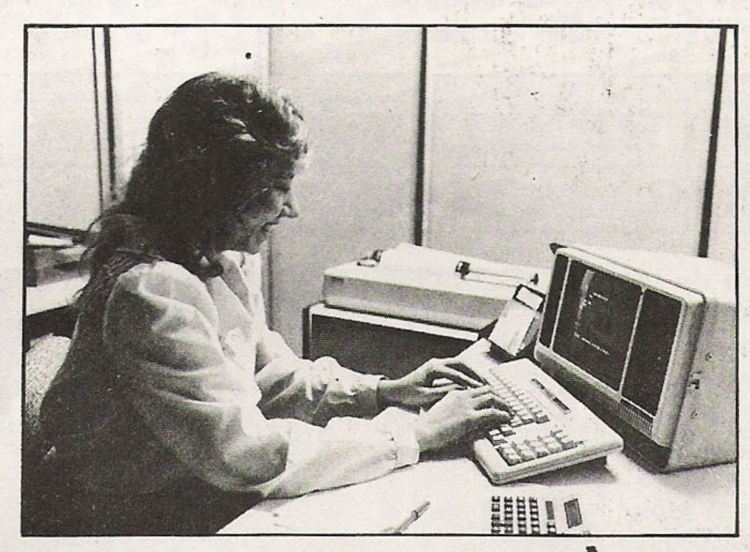
Distribuidor en Capital: Martino, Juan de Garay 358, P.B. Capital.
Distribuidor interior: DGP: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800.

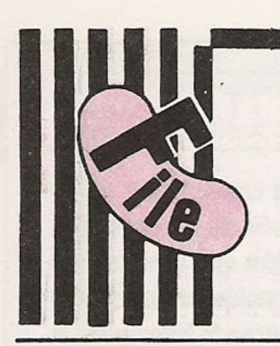
Sumario

BASES DE DATOS

Hace cientos de años, cuando el Inca era dueño y señor de gran parte de lo que hoy es Sudamérica, los europeos se admiraban del grado de información que tenía en relación a los hechos que sucedían en cualquier rincón del Imperio. Justamente, su poder residía en la información que manejaba y atesoraba.
Hoy, en los países desarrollados, se estima que quien posee "la información" tiene más posibilidades de dominar cualquier situación. Ya sea geo-política, social, cultural o económica.
Argentina ya comprendió cuales son las reglas del juego actuales. La comunicación informática o telemática es una realidad en nuestro país. Para ello, las bases de datos son las claves que permitirán entrar en el verdadero mundo de la información. En la actualidad, por medio de la computadora y el modem MSX podemos acceder a las más vastas redes mundiales de datos. Una verdadera transformación tecnológica está en marcha y nosotros podemos ser partícipes.

Los Editores





NUEVO SOFTWARE

PROSOFT, proveedor exclusivo del software para TALENT MSX pone a disposición de los usuarios una línea de casete, disquets y catridges con una gran variedad de programa.
En casetes se pueden encontrar juegos, soft educativos y utilitarios.
También en cartridge hay una amplia gama de utilitarios y entretenimientos.
La variedad de disquets también es grande. Van desde los sistemas de sueldos y jornales, contabilidad general, gestión de compras, gestión de ventas hasta los más divertidos juegos para el esparcimiento de los usuarios.
A continuación presentamos algunos de los programas lanzados al mercado. Utilitarios: Pascal, Ensamblador, Desensamblador y Ficheros Educativos: Física I, Inglés I y II, Cosmos y Países del Mundo. Juegos: F-16, Adventure Antartic; Frogger, Time Pilot, Pac Man, War head, Super Cobra, Pingüilandia, Jet Bomber, y otros tantos. Todos con muy buena definición de imágenes, excelente sonido y apasionante trama.

SINTETIZADOR

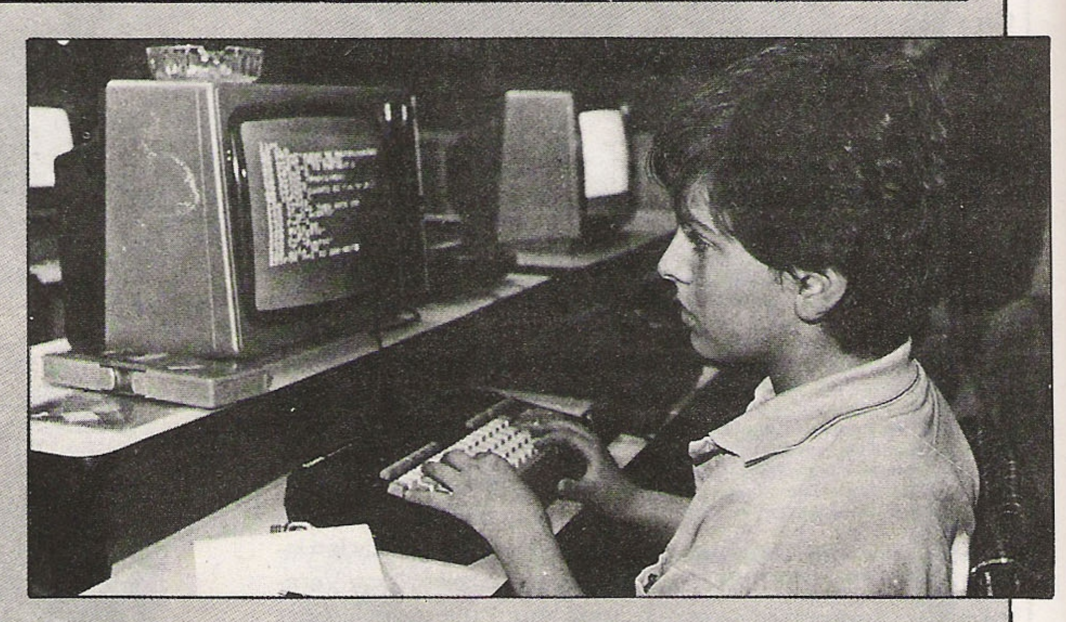
Desde que se desarrolló el MSX2, se ha destacado sobre todo su capacidad gráfica, pero su capacidad sonora resultaba del mismo nivel que MSX1.
Nos han dicho que esperáramos pacientemente una plaqueta llamada MSX-AUDIO y aquí está el primer prototipo.
Según cuentan los pajarillos, esta plaqueta tiene todas las capacidades para emular todo un sintetizador, con 8 canales diferenciales y 63 voces preprogramadas, de donde podríamos indicar:

Piano eléctrico
Piano acústico
Sintetizador tipo "Moog"
Arpa
Flauta
Trompetas
Violines
Batería
Guitarra
Organos
Bajo
etc.

También posee salida monoaural estéreo y capacidad de muestreo, o sea se puede tomar cualquier sonido de la naturaleza y digitalizarlo.
Además, su fuente sonora es en FM, por lo tanto, todo sonido que se encuentre en la naturaleza podrá ser imitado a través de esta plaqueta.
Por supuesto es compatible MSX o MSX2 y poseerá un teclado de piano adhoc para los amantes de la música.
También incorpora instrucciones expandidas para el MSX-BASIC, con lo que su programación se hace muy similar al macro-lenguaje de sonido que utiliza la instrucción PLAY del MSX-BASIC.

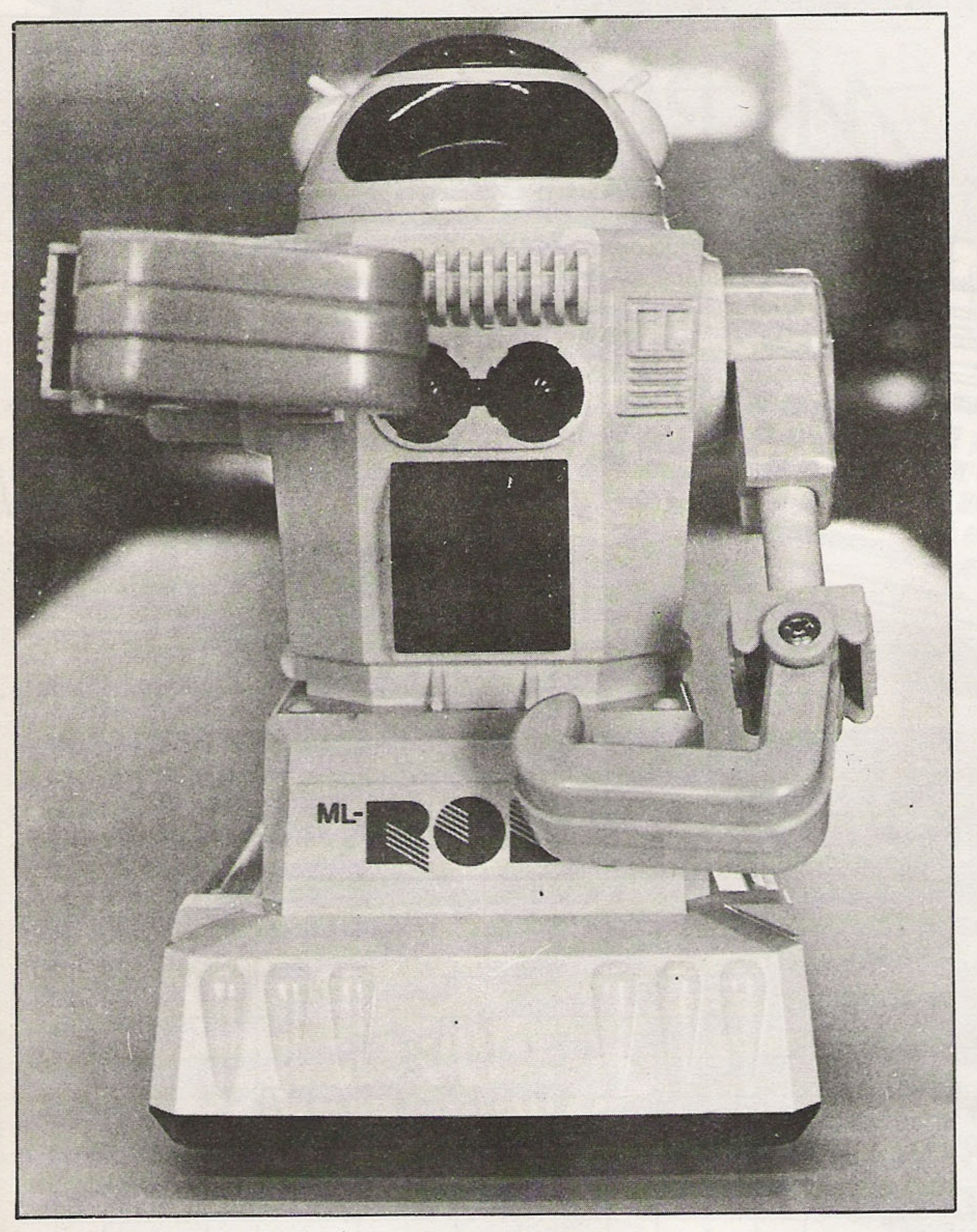
TELEMATICA Y LA INFORMATICA EDUCATIVA

La constante evolución de la aplicación de la informática en la Educación es una de las preocupaciones de Telemática. Esta empresa decidió incorporarse en el mercado educativo hacia fines del año pasado. Las alternativas en el momento no eran favorables. Por un lado aparecía la posibilidad de las costosas computadoras profesionales. Por el otro, las económicas home computers, que no satisfacían suficientemente los requerimientos educativos.
Telemática se propuso romper la disyuntiva ofreciendo una combinación para lograr una buena propuesta pedagógica. De este modo logró dotar laboratorios con una cantidad de equipos proporcional a 2 o 3 alumnos por máquina, cuya potencia posibilita una verdadera aplicación futura, y su costo es accesible.
TALENT tuvo una favorable respuesta por parte de sus destinatarios —los establecimientos educativos— no sólo de Capital Federal, sino del interior del país. Algunos de estos son: Escuela Argentina Modelo, Escuela Argentina 2000, Universidad del Salvador y la Facultad de Bioingeniería de la Provincia de Entre Ríos.



Esta gran demanda impulsó a Telemática "TALENT" a la formación del Departamento de Informática y Educación. El organismo brinda apoyo integral a Escuelas, Universidades y también a los distribuidores de todo el país.

EL ORIGINAL ROBOT MSX



No existen en el mercado poderosos seres electrónicos que realicen tareas complejas que ayuden al hombre. Pero sí hay compañías electrónicas que investigan constantemente en este campo y ésta es una aproximación a los futuros seres que habitarán nuestro mundo.

Las MSX han sido utilizadas para entretenernos con programas originales, producir gráficos y sonidos atrayentes. Pero, ¿se imaginan a nuestras computadoras haciendo el papel de "ama de llaves" y controlando a un regimiento de robots que nos sirvan el desayuno en la cama, nos alcancen los zapatos y nos traigan el diario?

No nos hagamos demasiadas fantasías, todavía esto sigue siendo un proyecto. Es indiscutible que los japoneses mantienen la delantera en este tema. Una compañía nipona expuso un robot que responde al nombre de ROBO. Este se arrastra por el suelo empujado por dos rueditas de movimientos individuales.

ROBO realiza tareas simples y todas de movimiento. Sin embargo, es sólo un completo "muñeco" que ayuda a la introducción de la era de los robots.

Por ahora no se vende en la Argentina, pero podemos encontrar uno en el club de usuarios MSX.

Está compuesto por dos brazos que carecen de movimiento (por eso todavía no tendremos el placer de que nos sirvan el desayuno).

ROBO es relativamente pequeño, mide alrededor de 19 cm. Por lo tanto es muy difícil que nos pueda proteger de los ladrones que invadan nuestro hogar.

Las alfombras no son obstáculo para estos "monstruitos". El desplazamiento sobre ellas es relativamente bueno en comparación con la mayoría de las máquinas de este tipo.

ROBO está formado por tres partes principales: el robot en sí, un transmisor de radio y software adecuado.

El transmisor es una pequeña caja blanca, alimentado por una pila y se conecta a la MSX en la ficha destinada a las impresoras.

El robot no posee sensores u otros dispositivos que necesiten ser conectados con la computadora.

El pórtico se deja libre para conectar la disketera u otro periférico.

La computadora envía las órdenes al transmisor por medio de señales digitales. Estas son captadas por la antena del transmisor y convertidas a señal analógica de radio de aproximadamente 27

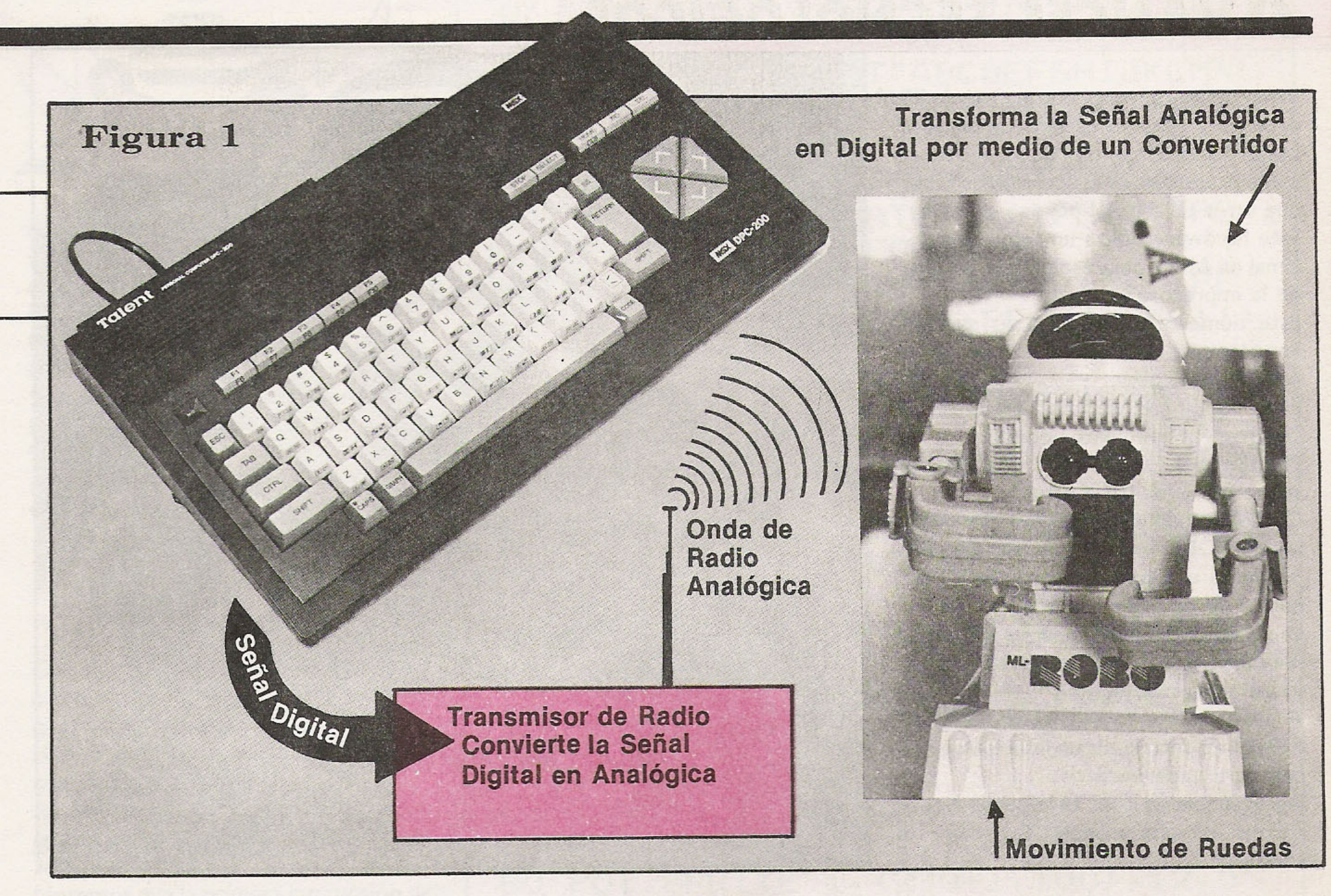


Figura 1

MHz de frecuencia. Las señales son enviadas en forma de ondas de radio y captadas por la antena de ROBO, que por medio de un conversor analógico/digital son codificadas y controlan los motores. Ver figura 1.

Es difícil decir si habrá problemas de interferencia. Podremos controlar a ROBO desde una distancia de aproximadamente 23 metros y atravesar algunas paredes sin problemas.

En cuanto al software, éste permite escoger controlar al robot a través del joystick o de un programa.

El cuerpo en sí utiliza dos ruedas con motores independientes que posibilitan ocho movimientos:

adelante, atrás,
adelante-derecha, adelante-izquierda,
atrás-derecha, atrás-izquierda,
giro-derecha y giro-izquierda,

que pueden ser ordenados individualmente a cada motor. También es posible permanecer quietos.

ROBO tiene motores eléctricos estandar, mientras que la mayoría de estos "aparatos" utilizan servomotor porque permiten varios tipos de soft controlador y son un poco más precisos. Estos últimos, sin embargo, tienen la desventaja de ser caros y consumir mucha energía.

Pero la desventaja de los motores es-

cia, dos motores de este tipo pueden dar diferentes respuestas a iguales impulsos voltaicos.

Por lo tanto si damos a los dos motores la orden de avanzar, uno puede ser más lerdo que el otro y provocar un giro simulando un arco en vez de moverse derecho hacia adelante.

Este inconveniente es resuelto agregándole un compensador de movimientos en la parte posterior del robot.

El programa que acompaña a este hardware, presenta la opción de grabar o cargar en cinta.

Si quisiéramos programar movimientos, esto se realiza entrando las iniciales de las direcciones. Se necesitan dos letras, una para cada motor, seguidas por un número que indica el tiempo en que se ejecutará dicha instrucción. Esta unidad está dada en décimas de segundos.

Se permiten entrar tres instrucciones por línea y hay espacio para siete líneas. Pero la última instrucción debe ser un "END" o sea EE, entonces solamente se pueden entrar 20 órdenes de movimientos.

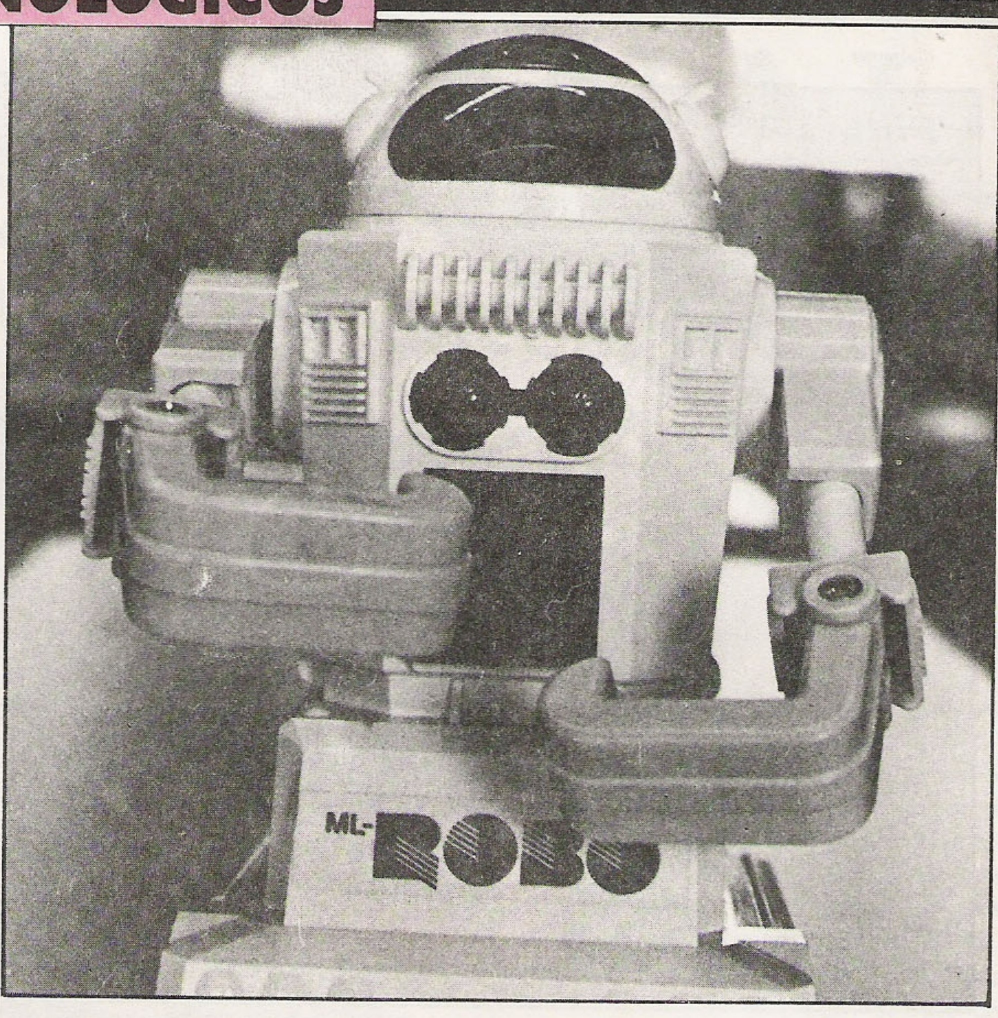
Por ejemplo: RR10FL20LL10 moverá hacia la derecha 1 segundo, en diagonal hacia arriba y a la izquierda 2 segundos y por último a la izquierda 1 segundo nuevamente.

El software que viene con ROBO no está protegido, o sea que podemos acceder al listado y ver cómo funciona o bien modificarlo. De esta manera se puede incorporar la opción de trabajar con disco en vez de solamente con cassettes. También sería aconsejable modificarlo

Figura 2 OUT &H91,n

n	Motor izquierdo	Motor derecho	Movimiento
0	Parado	Parado	Parado
1	Parado	Adelante	Adelante-Izquierda
2	Parado	Atrás	Atrás-Izquierda
3	Parado	Parado	Parado
4	Adelante	Parado	Adelante-Derecha
5	Adelante	Adelante	Adelante
6	Adelante	Atrás	Girar a la Derecha
7	Adelante	Parado	Adelante-Derecha
8	Atrás	Parado	Atrás-Derecha
9	Atrás	Adelante	Girar a la Izquierda
10	Atrás	Atrás	Atrás

para que nos permita entrar más de 20 órdenes y esto no es tan complicado.
Las órdenes de control de los motores son ejecutadas por los comandos OUT que envía un byte al pórtico de salida.
Este hardware utiliza un valor hexadecimal de &H91 para especificar la salida de la impresora.
Este número hexadecimal es seguido por otro que indica el movimiento para desplazar el robot
En la figura 2 mostramos una tabla de códigos y movimientos posibles.
Por ejemplo OUT &H91,10 ordena al robot moverse hacia atrás.
Como se pudo ver, crear los movimientos de ROBO es algo muy sencillo, que permitirá su desplazamiento por los espacios de las oficinas y hogares pero a veces no podremos visualizar sus movimientos por esconderse detrás o debajo de algún mueble.
A pesar de ser solamente un juguete, ROBO no deja de ser un interesante robot para experimentos.
Agregándole un lápiz, se lo puede transformar en una especie de mouse o tortuga.





MELODIA

Este es un pequeño programa que al ejecutarlo hará sonar una agradable melodía. Hay veces que realizamos programas casi completos pero nos faltaría agregarles alguna música para hacerlos más entretenidos.
Aquí les proponemos una, a la que también le podrán hacer algunos cambios con el fin de darle un toque propio.

```
10 ' ******** MELODIA ********
20 ' COMIENZA A COPIAR DESDE LA
LINEA 30
30 A$="R4.O5L8CFCFGAB-16A16GA16B
-16O6CO5F16G16A16G16F16E16F4.CFCF
GAB-16A16GA16B-16O6CO5F16G16A16G1
6F16E16F4.O6CC.D16CO5B-AB-O6C"
40 B$="O6L8CDDL16CO5B-AGF4L8ECFE
16F16GF16G16AG16A16B-A16B-16O6CO5
F16G16A16G16F16E16F4."
50 PLAY A$: PLAY B$: GOTO 50
```

En la línea 50, la última sentencia se puede borrar quedando:
50 PLAY A$: PLAY B$
Otro efecto sería agregar en esta línea otra sentencia PLAY para que quede de la siguiente manera:
50 PLAY A$: PLAY B$: PLAY B$
Así como estas dos variantes que hemos propuesto, existen muchas otras, que con sólo modificar una pequeña parte del programa, obtendremos diferentes resultados.



CARACTERES MAGICOS

Para crear nuevos sets de caracteres, muy útiles para algunos programas, proponemos el programita de la figura.
Cuando hayamos terminado de copiar y verificar el listado, es conveniente grabarlo antes de hacerlo correr.
Al ejecutar el programa, la pantalla irá modificándose, produciendo efectos interesantes.
Cuando estemos satisfechos con alguna trama, presionemos cualquier tecla.
Aparecerán en la pantalla dos direcciones: 62399! y 62401! cada una con un valor.
Si en cualquier programa introducimos las sentencias:
POKE 62399,JJ
POKE 62401,KK





cuentren en SCREEN 1. JJ y KK son los valores que aparecen en pantalla al lado de cada dirección.
No olvidemos que solamente se modifican los caracteres si trabajamos en el modo de pantalla 1.
Si por ejemplo ingresamos las sentencias: POKE 62399!,O: POKE 62401!,28: SCREEN1
modificará los caracteres y dibujará un fondo original.

```
10 KEY OFF:
20 FOR Q=0 TO 255:FOR W=0 TO 255
30 POKE 62399!,Q:POKE 62401!,W
40 SCREEN 1:PRINT"ABCDEFGHIJKLMN
OPQRSTUVWXYZ":FOR F=1 TO 300:NEXT
F
50 IF INKEY$="" THEN NEXT W,Q
60 SCREEN 0: LOCATE 2,6:PRINT"PO
KE COLOR","POKE FORMA"
70 LOCATE 0,8:PRINT 62399!",";PE
EK(62399!),62401!",";PEEK(62401!)
:KEY ON
80 LOCATE 0,14:PRINT"Presione RE
TURN para continuar         y CRTL/
STOP para parar:":INPUT C$:NEXT W
,Q
```

CHOQUE DE SPRITES

Con el listado de la figura , veremos de qué forma se utiliza la detección de choques entre sprites.
Hemos definido dos sprites, cada uno con un color diferente.
Al detectarse una superposición entre ambos gráficos, se ejecutarán las líneas 110, 120 y 130.
Notemos que la primera instrucción de la rutina de choque es: SPRITE OFF. Luego escribimos los efectos que queremos que se realicen y por último volvemos a abrir el control de posición de sprites con SPRITE ON.
Veamos qué sucede si provocamos pequeños cambios como:

- borrar la línea 110 y modificar la 10 por 10 ON SPRITE GOSUB 120
- borrar la primera sentencia de la línea 130 para quedar 130 RETURN

o bien modificar las consecuencias ante la detección de una superposición de sprites.

```
10 ON SPRITE GOSUB 110
20 SCREEN 2,1
30 SPRITE$(0)=STRING$(8,CHR$(255
))
40 SPRITE$(1)=STRING$(8,CHR$(255
))
50 SPRITE ON
60 FOR I=10 TO 240
70 PUT SPRITE 0,(I,100),6,0
80 PUT SPRITE 1,(230-I,100),12,0
90 NEXT
100 GOTO 50
110 SPRITE OFF
120 PUT SPRITE 0,(I,100),10:BEEP
130 SPRITE ON:RETURN
```

ESPIRALES

Este truco lo envió un lector llamado Ricardo Pesce y es una forma muy sencilla de producir efectos gráficos agradables. Te felicitamos Ricardo por haber conseguido estos resultados con una impecable sencillez.
Sugerimos agregar una línea:

```
115 CO=INT(RND(1)*13+2)
```

y reemplazar el número 15 de la línea 130 por CO. Entonces la 130 quedará:

```
130 LINE -(X,Y),CO
```

de esta manera le añadimos colores aleatorios al gráfico propuesto por Ricardo.

```
10 ' DIBUJA ESPIRALES ARQUIMEDEA
NAS.
20 ' INGRESANDO "0" SE FINALIZA
30 ' ALGUNOS VALORES INTERESANTE
S SON: 60,61,72,73,90,91,120,121
40 CLS
50 INPUT "GRADOS: ";D
60 IF D=0 THEN CLS: END
70 SCREEN 2
80 COLOR 15,1,1:CLS
90 D=D/57.29578#
100 PSET(128,92),15
110 FOR R=0 TO 100 STEP D
120 X=R*COS(R):X=X+128:Y=R*SIN(R
):Y=Y*.7+92
130 LINE -(X,Y),15
140 NEXT R
150 FOR Z=1 TO 2000: NEXT
160 SCREEN 0: COLOR 15,4,4
170 GOTO 50
```

TRANSFORMACION DE NUMEROS

Desde primer grado hemos aprendido a operar con los números en sistema decimal, pero ahora la computadora nos introduce a otra forma de expresar los números fundamentalmente en base dos.

Cuando aún no estaban implantadas las pautas del lenguaje de los hombres primitivos, tampoco existían los números. Pero intuitivamente pensamos que las cantidades siempre existieron como conjunto, pudiéndose notar siempre qué conjunto abarcaba mayor cantidad de elementos.

Pero a estos homosapiens les hizo falta saber con mayor precisión la "cantidad" de elementos que componían un conjunto en cuestión.

Supongamos que en determinado día, un hombre primitivo pescó lo que para nosotros significa "muchos" peces. Pero se encontró con que en días posteriores su pesca creció. Para un lenguaje sencillo supongamos que la cantidad de peces cazados era "un poco más que mucho". A medida que su caza aumentaba, saber la cantidad de pescados obtenidos se hacía más complicado. Entonces surgió la necesidad de crear "algo" para que el hombre pudiera ordenar y clasificar todas las percepciones de sus sentidos, dando orígen al que hoy llamamos "sistema de numeración". Es lógico pensar que el sistema de numeración de las distintas civilizaciones ha ido cambiando a través del tiempo.

Por ejemplo, Babilonia antigua manejaba el sistema sexagesimal (base 60). El orígen y el por qué los babilonios hayan adoptado este sistema no es seguro. Los historiadores sostienen que se debe a la fusión de dos civilizaciones, una usaba el sistema senario (base 6), mientras que la otra trabajaba con el sistema decimal (base 10). La cultura de este pueblo aún perdura pues las mediciones de horas, minutos y segundos sigue siendo en base sexagesimal (1h=60m; 1m=60s) y también se utiliza para medir ángulos (1° =60'; 1'=60").

Los aztecas y mayas emplearon el sistema vigesimal (base 20). Esta costumbre apenas subsiste en el idioma francés, por ejemplo 80 se dice "quatre-vingt" o sea "cuatro veces veinte".

Otro sistema era el duodecimal (base 12). Su orígen está ligado al cálculo numérico con los dedos, pues los cuatro dedos de la mano (índice, mayor, anular y meñique) tienen cada uno tres falanges que hacen un total de 12. Entonces se puede contar de 1 hasta 12 pasando el pulgar por las falanges quedando doce como unidad de este sistema. Aún se utiliza este método de conteo y en lugar de "doce" lo llamamos "docena". Con esta unidad se clasifican varios elementos como cuchillos y huevos.

Pero otras civilizaciones contaban con los dedos de las manos y los posibles números a obtener (contando una sola vez cada mano) son del 1 al 10. A este sistema se lo llamó decimal y es el que perdura aún.

En computación es muy utilizado el sistema binario (base 2), pero antes de explicar este método veamos el sistema que usamos habitualmente.

Esta nota será explicada de manera que pueda ser entendida por todos los lectores, por eso pedimos disculpas a aquellos que se encuentran avanzados en este tema porque tal vez caigamos en la redundancia de explicar algunas propiedades de los números.

Figura 1

$96=6x10^0+9x10^1$
$395=5x10^0+9x10^1+3x10^2$
$1904=4x10^0+0x10^1+9x10^2+1x10^3$
$20483=3x10^0+8x10^1+4x10^2+0x10^3+2x10^4$

En la escuela primaria nos enseñaron que, por ejemplo, en el número 486 el 6 ocupa el lugar de las unidades, el 8 de las decenas y el 4 el de las centenas. Esto significa que 486 se escribe: 6x1+8x10+4x100 (1). Pero $10^0=1$; $10^1=10$ y $10^2=100$ entonces reemplazando estas nuevas expresiones en (1), veremos que $486=6x10^0+8x10^1+4x10^2$.

Observemos entonces que multiplicamos todos los dígitos, comenzando desde el extremo derecho, por diez elevado a una potencia. El seis ocupa la primera posición (a partir del extremo derecho) y está multiplicado por diez elevado a la potencia cero; el ocho ocupa el segundo lugar y está multiplicado por diez elevado a la primera potencia y por último, cuatro está en el tercer lugar y lo multiplica diez elevado al cuadrado. Como conclusión de esto podemos decir que los dígitos están multiplicados por 10 elevado a la posición del dígito menos uno. Volvamos a escribir 486 de otra forma pero incorporando lo que acabamos de explicar: $486= 6x10^{(1-1)}+8x10^{(2-1)}+4x10^{(3-1)}$. Recordemos que una de las propiedades de la potenciación es que cualquier número elevado a la potencia cero es igual a uno y en particular $10^0=1$.

Generalicemos todo lo explicado diciendo que todo número natural se puede escribir en cualquier sistema de numeración, multiplicando sus dígitos por la base del sistema al que se quiere llegar, elevada por la posición de dicho dígito menos 1. Si esta definición no nos parece clara, volvamos a leer el párrafo anterior. Veamos algunos ejemplos en la figura 1.

Al dividir un número por diez, los posibles restos (miremos la figura 2 si no nos acordamos a qué se le llama resto), son "0123456789" pero nunca un número mayor o igual a 10.

En el sistema decimal, los números se escriben combinando los dígitos "012 3456789" que coinciden con los posibles restos. Esta coincidencia no es casualidad pues para pasar un número escrito en base 10 a otro sistema, hay que dividir sucesivamente hasta que el resto sea menor que el divisor.

En la figura 3 hay un ejemplo de esta división.

Pero en computación se utiliza otro sistema de numeración. Pues resultó más fácil diseñar dispositivos electrónicos que distingan dos estados —si hay o no un pulso electrónico- -, entonces nos bastó contar con dos dígitos: 0 y 1. Estos dígitos corresponden a los posibles restos de dividir por dos (el

resto de dividir por un número determinado nunca puede superar o igualar al divisor), entonces debido a esto se trabaja en computación con el sistema binario.

Para pasar un número decimal a binario hay que dividir el número a convertir sucesivamente por dos. Si queremos pasar el número 23 a binario, habría que dividir el 23 sucesivamente por la base del sistema que vendría a ser 2. Luego se toma el último cociente y todos los restos comenzando por el último obtenido. Con estos dígitos puestos uno al lado del otro, de izquierda a derecha, se obtiene el número 23 convertido a la base dos.

En la figura 4 puede observarse como se realizan las operaciones explicadas.

De esta forma, cualquier entero al dividirlo por una base, se lo pasa al sistema de dicha base. ¿Pero cómo haríamos si tuviéramos un número binario y lo quisiéramos pasar al decimal?. Para esto hay que tener en cuenta lo que explicamos más arriba, la definición delirante: todo número natural expresado en alguna base se puede pasar a cualquier otro sistema si multiplicamos cada uno de sus dígitos por la base elevada por la posición de dicho dígito menos uno. Esto, en símbolos, está representado en la figura 5.

¿Pero cómo podemos averiguar el significado decimal de un número expresado en binario?

Por ejemplo, supongamos que nuestra inquietud es averiguar el valor decimal de 1101001 (éste está dado en binario). Según la definición anterior, $1101001=1x2^0+0x2^1+0x2^2+1x2^3+0x2^4+1x2^5+1x2^6$ y realizando las operaciones resulta $1101001_2=105_{10}$ (los subíndices indican en qué sistema están escritas las dos cantidades).

La operación más sencilla de realizar en este sistema es la adición. Recomendamos operar solamente con dos cantidades hasta familiarizarnos con el manejo de números en esta base.

Elijamos dos cantidades binarias al azar, por ejemplo 101_2 (5_{10}) y 1010_2 (10_{10}). Para sumarlas primero hay que igualar la cantidad de dígitos, agregando ceros a la izquierda. En nuestro ejemplo, un 101_2 tiene tres dígitos y 1010_2 cuatro. Esto implica que deberemos completar 101_2 con un cero a la izquierda: 0101_2. Ahora sí ya podemos sumarlos encolumnándolos

```
  0101
+ 1010
```

Igual que la suma en decimal, comenzamos sumando de derecha a izquierda. En consecuencia nos queda la operación 1+0=1, la segunda columna es 0+1=1, la tercera 1+0=1 y así siguiendo se llega:

Figura 2

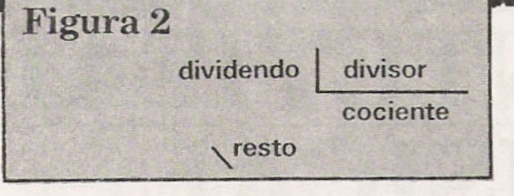


Figura 3

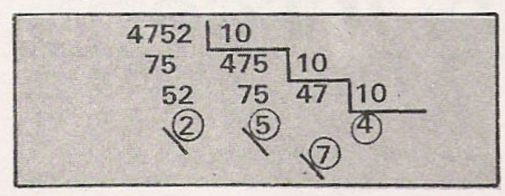


Figura 4

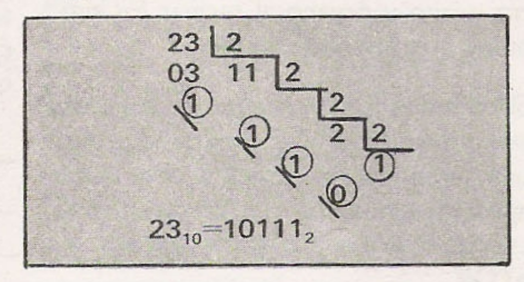


Figura 5

P = base a la que se quiere pasar pasar el número

ABCDE = número de 5 dígitos escrito en cualquier base

i=lugar que ocupa el dígito, contando desde la derecha

A B C D E

i=5 i=4 i=3 i=2 i=1

$ABCDE=ExP^{i-1}+DxP^{i-1}+CxP^{i-1}+BxP^{i-1}+AxP^{i-1}$

Figura 6

```
 10010      01111     1110
+01011     +10100    +0111
 11101     100011    10101
```

Figura 7

DECIMALES	BINARIO
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

```
  0101
+ 1010
  1101₂
```

Pero no en todas las adiciones contamos con cifras que efectúan una operación sencilla como el ejemplo que vimos.

Supongamos que debemos adicionar las cifras 1010_2 y 111_2. Primero volvemos a completar con cero hasta que las cantidades tengan el mismo número de dígitos.

```
  1010
+ 0111
```

La primera columna es 0+1=0; 1+1=2. pero recordemos que en base 2 el "0" y el "1" son los únicos dígitos representativos, en consecuencia hay que pasar el resultado de esta suma a binario: $1+1=2_{10}=10_2$. Este último número tiene dos dígitos, entonces no se puede poner como resultado de la siguiente columna. Pero si separamos los dos dígitos, el "0" va como resultado de la segunda columna y el 1 se suma a la tercera.

```
  1010
+ 0111
   1
   00
```

Lo mismo pasa con la columna 3; tenemos que sumar $0+1+1=2_{10}=10_2$, el dígito de la derecha queda como resultado de la columna que sumemos y el dígito de la izquierda pasa sumando en la columna siguiente.

```
  1010
  0111
+ 1
 10001₂
```

Pasemos las tres cantidades al sistema decimal y veamos si la suma está correctamente realizada: $1010_2=10_{10}$; $0111=7_{10}$ y $10001_2=17_{10}$

En la figura 6 hay algunos ejemplos de adición.

La palabra BIT proviene del inglés Binary digIT (dígito binario). Cada dígito entonces ocupa un bit. El número más grande que se puede representar con dos bits es $11_2=3$; con tres bits es $111_o=7_{10}$ y con cuatro bits es $1111_2=15_{10}$. En la figura 7 están representados los valores posibles en decimal y binario de los números entre "0" y "15".

Si n es la cantidad máxima de dígitos de un número binario, el valor más alto que se puede obtener en decimal está dado por la expresión: 2^n-1. Por ejemplo, si reemplazamos en la fórmula con n=4, nos queda $2^4-1=15$ el máximo valor de cuatro bits representable como habíamos visto antes.

Pero la computadora MSX, también trabaja con sistema hexadecimal y octal (base 16 y 8 respectivamente). Estas formas de aritmética las veremos en el siguiente número y completaremos las operaciones entre los sistemas.



SINTETIZADOR MUSICAL

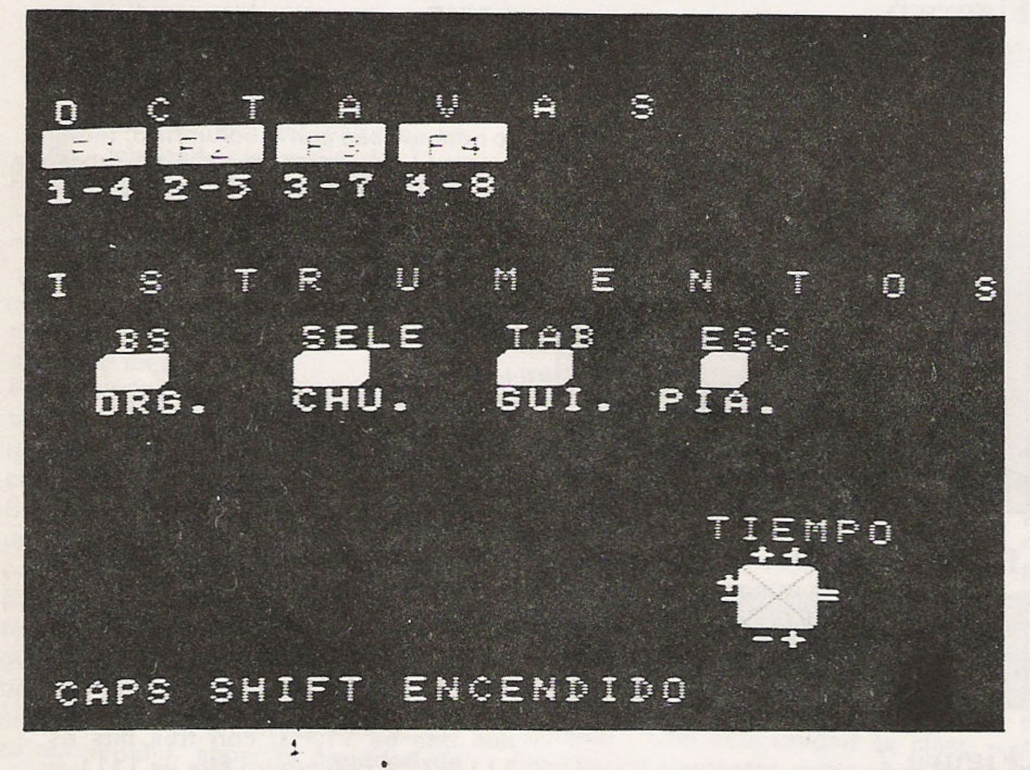
CLASE: JUEGO

Con este programa podremos generar notas musicales en cualquier de las ocho octavas disponibles.

El tiempo de duración de cada nota se cambia pulsando los cursores. Por ejemplo, presionando la flecha que indica hacia arriba, provocará que la nota dure mucho tiempo.

Otra opción con que cuenta este programa es que se transformará en guitarra, piano u órgano al teclear los símbolos que los representan.

Si detenemos la ejecución de las notas por medio del STOP, podremos tocar una secuencia de treinta y siete notas aproximadamente, que se escucharán cuando volvamos a tocar la tecla de STOP.



Al cortarse el programa, el color de las letras será igual que el de fondo, pero a no desesperarse y entremos la instrucción COLOR 11,1,1 o cualquier otra combinación de colores.

VARIABLES IMPORTANTES

A$ Tecla presionada

ESTRUCTURA DEL PROGRAMA:

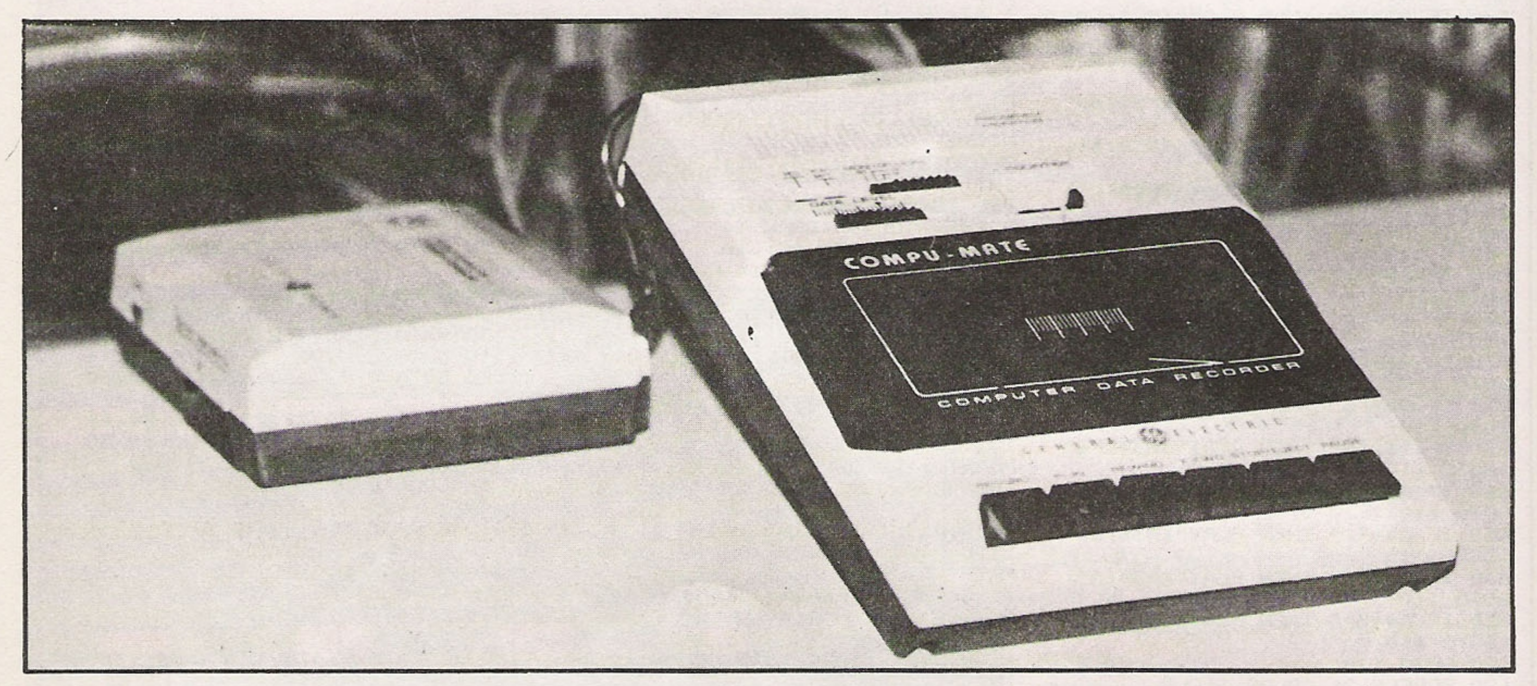
40: inicializa variables de la instrucción PLAY
50—60: destina comandos para las teclas de función
70—100: destina octavas para cada función
110: acepta tecla
120—150: efectos para cada instrumento
160—170: modifica el tiempo de ejecución de las notas
200—670: suena la nota elegida
690—11260: grafica pantalla
1170—1540: modifica el set de caracteres
1550—1660: pantalla de explicación

```
10 GOSUB 1170
20 GOSUB 690
30 '*******************************
****
40 PLAY"T255L4"
50 COLOR 1,1:FOR F=1 TO 5:KEY(F)
ON:NEXT
60 ON KEY GOSUB 70,80,90,100:GOT
O 110
70 M$="O1":M1$="O2":M2$="O3":M3$
="O4":M4$="O5":RETURN
80 M$="O2":M1$="O3":M2$="O4":M3$
="O5":M4$="O6":RETURN
90 M$="O3":M1$="O4":M2$="O5":M3$
="O6":M4$="O7":RETURN
100 M$="O4":M1$="O5":M2$="O6":M3
$="O7":M4$="O8":RETURN
110 A$=INKEY$
120 IF A$=CHR$(8) THEN PLAY"V6S9
M65535L3":GOTO 110
130 IF A$=CHR$(27) THEN PLAY "V5
S9M65535L4":GOTO 110
140 IF A$=CHR$(9) THEN PLAY "V9S
1M2000":GOTO 110
150 IF A$=CHR$(24) THEN PLAY "V1
4S4M300":GOTO 110
160 IF A$=CHR$(30) THEN PLAY "T8
0":GOTO 110
170 IF A$=CHR$(29) THEN PLAY "T1
50":GOTO 110
180 IF A$=CHR$(31) THEN PLAY "T2
00":GOTO 110
190 IF A$=CHR$(28) THEN PLAY "T2
55":GOTO 110
200 IF A$="" THEN 110
210 IF A$="1" THEN PLAY"XM$;C":G
OTO 110
220 IF A$="2" THEN PLAY"XM$;C#":
GOTO 110
230 IF A$="3" THEN PLAY"XM$;D":G
OTO 110
240 IF A$="4" THEN PLAY"XM$;D#":
GOTO 110
250 IF A$="5" THEN PLAY"XM$;E":G
OTO 110
260 IF A$="6" THEN PLAY"XM$;F":G
OTO 110
270 IF A$="7" THEN PLAY"XM$;F#":
GOTO 110
280 IF A$="8" THEN PLAY"XM$;G":G
OTO 110
290 IF A$="9" THEN PLAY"XM$;G#":
GOTO 110
300 IF A$="0" THEN PLAY"XM$;A":G
OTO 110
310 IF A$="-" THEN PLAY"XM$;A#":
GOTO 110
320 IF A$="=" THEN PLAY"XM$;B":G
OTO 110
330 IF A$="\" THEN PLAY"XM1$;C":
GOTO 110
340 IF A$="Q" THEN PLAY"XM1$;C#"
:GOTO 110
350 IF A$="W" THEN PLAY"XM1$;D":
GOTO 110
360 IF A$="E" THEN PLAY"XM1$;D#"
:GOTO 110
370 IF A$="R" THEN PLAY"XM1$;E":
GOTO 110
380 IF A$="T" THEN PLAY"XM1$;F":
GOTO 110
390 IF A$="Y" THEN PLAY"XM1$;F#"
:GOTO 110
```

```
400 IF A$="U" THEN PLAY"XM1$;G":
GOTO 110
410 IF A$="I" THEN PLAY"XM1$;G#"
:GOTO 110
420 IF A$="O" THEN PLAY"XM1$;A":
GOTO 110
430 IF A$="P" THEN PLAY"XM1$;A#"
:GOTO 110
440 IF A$="[" THEN PLAY"XM1$;B":
GOTO 110
450 IF A$="]" THEN PLAY"XM2$;C":
GOTO 110
460 IF A$="A" THEN PLAY"XM2$;C#"
:GOTO 110
470 IF A$="S" THEN PLAY"XM2$;D":
GOTO 110
480 IF A$="D" THEN PLAY"XM2$;D#"
:GOTO 110
490 IF A$="F" THEN PLAY"XM2$;E":
GOTO 110
500 IF A$="G" THEN PLAY"XM2$;F":
GOTO 110
510 IF A$="H" THEN PLAY"XM2$;F#"
:GOTO 110
520 IF A$="J" THEN PLAY"XM2$;G":
GOTO 110
530 IF A$="K" THEN PLAY"XM2$;G#"
:GOTO 110
540 IF A$="L" THEN PLAY"XM2$;A":
GOTO 110
550 IF A$=";" THEN PLAY"XM2$;A#"
:GOTO 110
560 IF A$="'" THEN PLAY"XM2$;B":
GOTO 110
570 IF A$="`" THEN PLAY"XM3$;C":
GOTO 110
580 IF A$="Z" THEN PLAY"XM3$;C#"
:GOTO 110
590 IF A$="X" THEN PLAY"XM3$;D":
GOTO 110
600 IF A$="C" THEN PLAY"XM3$;D#"
:GOTO 110
610 IF A$="V" THEN PLAY"XM3$;E":
GOTO 110
620 IF A$="B" THEN PLAY"XM3$;F":
GOTO 110
630 IF A$="N" THEN PLAY"XM3$;F#"
:GOTO 110
640 IF A$="M" THEN PLAY"XM3$;G":
GOTO 110
650 IF A$="," THEN PLAY"XM3$;G#"
:GOTO 110
660 IF A$="." THEN PLAY"XM3$;A":
GOTO 110
670 IF A$="/" THEN PLAY"XM3$;A#"
:GOTO 110
680 GOTO 110
690 SCREEN 2: COLOR,1:CLS
700 OPEN"GRP:"AS#1
710 DR$="S5H15R15G15U15F15U15G15
R15"
720 PSET (208,165):DRAW"C14A0":D
RAW"XDR$;"
730 PAINT (206,160),14
740 PAINT (200,164),14
750 PAINT (193,160),14
760 PAINT (200,153),14
770 PSET(208,165):DRAW"C1S5H15BR
15G15"
780 PSET (193,140),1:COLOR 11: P
RINT #1,"++"
790 PSET (193,166),1:COLOR 11: P
RINT #1,"-+"
800 PSET (209,152),1:COLOR 11: P
RINT #1,"="
810 PSET (185,149),1:COLOR 11: P
RINT #1,"+"
820 PSET (185,154),1:COLOR 11: P
RINT #1,"-"
830 PSET (182,132),15:COLOR 4: P
RINT #1,"TIEMPO"
840 PSET (20,182),15: PRINT #1,"
CAPS SHIFT ENCENDIDO"
850 RESTORE 890:FOR A=1 TO 4
860 READ PS,PT,PR,PE
870 LINE (PS,PT)-(PR,PE),11,BF
880 NEXT
890 DATA 15,10,40,20,45,10,70,20
,75,10,100,20,105,10,130,20
900 PSET (17,25),15:COLOR 9: PRI
NT #1,"1-4"
910 PSET (47,25),15:COLOR 9: PRI
NT #1,"2-5"
920 PSET (77,25),15:COLOR 9: PRI
NT #1,"3-7"
930 PSET (107,25),15:COLOR 9: PR
INT #1,"4-8"
940 PSET (18,1),15:COLOR 4: PRIN
T #1,"O C T A V A S"
950 PSET (22,13),11:COLOR 1: PRI
NT #1,"F1"
960 PSET (50,13),11:COLOR 1: PRI
NT #1,"F2"
970 PSET (82,13),11:COLOR 1: PRI
NT #1,"F3"
980 PSET (112,13),11:COLOR 1: PR
INT #1,"F4"
990 T$="A0C14R12E2U7L12G2D7":DRA
W"BM30,90C14":DRAW T$
1000 PAINT (35,85),14
1010 PSET (35,72),14:COLOR 12: P
RINT #1,"BS"
1020 DRAW"BM80,90C14":DRAW T$
1030 PAINT (85,85),14
1040 PSET (82,72),14:COLOR 12: P
RINT #1,"SELE"
1050 DRAW"BM130,90C14":DRAW T$
1060 PAINT (135,85),14
1070 PSET (132,72),14:COLOR 12:
PRINT #1,"TAB"
1080 DRAW"BM180,90C14R7E2U7L7G2D
7"
1090 PAINT (185,85),14
1100 PSET (180,72),14:COLOR 12:
PRINT #1,"ESC"
1110 PSET (17,54),15:COLOR 4: PR
INT #1,"I S T R U M E N T
 O S"
1120 PSET (30,92),15:COLOR 9: PR
INT #1,"ORG."
1130 PSET (80,92),15:COLOR 9: PR
INT #1,"CHU."
1140 PSET (130,92),15:COLOR 9: P
RINT #1,"GUI."
1150 PSET (170,92),15:COLOR 9: P
RINT #1,"PIA."
1160 RETURN
1170 COLOR 1,1: SCREEN 1
1180 CLS: KEY OFF
1190 RESTORE 1310
1200 FOR I=520 TO 631
1210 READ ZQ
1220 VPOKE I,ZQ
1230 NEXT I
1240 FOR I=1320 TO 1327
1250 READ ZQ
1260 VPOKE I,ZQ
1270 NEXT I
1280 FOR I=632 TO 727
1290 READ ZQ: VPOKE I,ZQ
1300 NEXT
1310 DATA 0,12,18,18,30,82,162,1
93
1320 DATA 0,182,68,68,126,65,65,
190
1330 DATA 0,56,84,136,128,130,68
,56
1340 DATA 0,120,132,66,66,68,120
,128
1350 DATA 0,184,68,64,112,64,68,
184
1360 DATA 0,126,137,72,14,8,16,9
6
1370 DATA 0,56,68,128,142,130,68
,56
1380 DATA 0,67,165,36,60,36,165,
66
1390 DATA 0,16,40,8,8,72,72,48
1400 DATA 0,60,68,132,4,68,72,48
1410 DATA 0,132,74,80,96,80,72,1
32
1420 DATA 0,40,88,40,72,8,8,15
1430 DATA 0,34,54,42,34,34,162,6
5
1440 DATA 0,34,50,42,38,34,162,6
4,8,34,50,42,38,34,162,64
1450 DATA 0,56,68,68,68,68,68,56
,56,68,130,98,60,32,32,32,56,68,6
8,68,84,72,52,0
1460 DATA 0,156,98,34,60,40,36,3
4
1470 DATA 0,56,68,72,56,4,132,12
0
1480 DATA 0,126,137,72,8,8,8,8
1490 DATA 0,32,82,18,18,18,18,12
1500 DATA 0,66,162,34,34,34,20,8
1510 DATA 0,0,0,0,0,0,0,0
1520 DATA 0,72,180,48,48,48,180,
72
1530 DATA 0,72,72,72,56,8,40,16
1540 DATA 0,60,66,20,8,20,32,30
1550 CLS: COLOR 11,1,1
1560 LOCATE 0,0: PRINT"«««««««««
««««««««««««««««««««"
1570 LOCATE 0,1:PRINT"«
                    «"
1580 LOCATE 0,2:PRINT"«     GENER
ADOR DE MUSICA      «"
1590 LOCATE 0,3:PRINT"«
                    «"
1600 LOCATE 0,4:PRINT"««««««««««
««««««««««««««««««««"
1610 LOCATE 0,7:PRINT"EL TIEMPO
SE CONTROLA CON LASTECLAS DE CURS
OR."
1620 LOCATE 0,11:PRINT"LAS OCTAV
AS SE ELIGEN CON LASTECLAS DE FUN
CION"
1630 LOCATE 0,15:PRINT"EL INSTRU
MENTO SE ELIGE PRE- SIONANDO LA T
ECLA INDICADA"
1640 LOCATE 0,20:PRINT"      PUL
SAR UNA TECLA"
1650 IF INKEY$="" THEN 1650
1660 RETURN
1670 END
```

CINTAS Y GRABADORES AL DESCUBIERTO

Los misterios que creemos que existen entre las computadoras y grabadores no son tan inexplicables ni hace falta tener conocimientos sólidos de física o de electrónica. Por eso aquí proponemos terminar con las suposiciones en torno a este tema.



La forma más económica de guardar datos para las operaciones de las computadoras es indudablemente utilizar cassettes. Estas cintas se convierten en la memoria auxiliar de las máquinas.

Aquí recurren la mayoría de los HOME COMPUTERS para leer información que luego se encargarán de procesar y una vez terminada esta tarea, es común que vuelvan a guardar los datos nuevos en este tipo de cintas.

Cuántos nos hemos sorprendido al ver por primera vez a una computadora "charlando" con un grabador. No entendíamos cómo aquél artefacto se convertía desde un intérprete de orquesta a una misteriosa memoria de la computadora seudopensante.

Pero nuestro razonamiento no superaba el dilema y nos resignamos a aceptar el nuevo hecho.

En esta nota nos hemos propuesto aclarar algunos puntos de este tema para ayudarnos a entender cuál es el misterio que une la computadora y el grabado.

¿Qué es una cinta?

Las cintas de los cassettes son fabricadas con plástico de larga vida, recubiertas por una delgada película de hierro. Las materias que componen esta cintas tienen la propiedad de permanecer magnetizadas, a través de un imán, por un período de tiempo relativamente extenso.

Imaginemos las cintas formadas por gran cantidad de dominios, campos magnéticos desorganizados que provocan una magnetización nula, producto de que cada partícula tiende a anular el efecto magnético de las otras.

En la figura 1 está representada una cinta desmagnetizada.

Pero al pasar una porción de cinta cerca de un imán, provocará que las partículas de hierro tomen todas la misma dirección como mostramos en la figura 2. De la misma forma se puede alinear una cantidad determinada de partículas. Esto depende de la intensidad de fuerza que genere el imán. Es decir que si la fuerza del campo magnético es débil, entonces serán pocas las moléculas que se alineen.

El imán, en un grabador, lo realiza el cabezal donde hay un electroimán que produce el campo magnético provocado por la corriente que lo atraviesa.

El cabezal del grabador está formado por un anillo magnético con una abertura pequeña. La figura 3 es un esquema del campo magnético que produce el cabezal del grabador.

En la cinta se efectuarán diferentes diseños magnéticos que dependen de la señal enviada por el cabezal.

El tamaño de la abertura puede variar, cuanto mayor sea ésta, mayor será la cantidad de cinta que magnetice.

En consecuencia, la rapidez para grabar una cinta depende de tres motivos:

1— abertura del cabezal de grabación
2— velocidad de arrastre de la cinta
3— calidad de la cinta.

¿Qué significan los ruidos extraños?

El único lenguaje que entiende la com-

putadora es el basado en el sistema binario que consiste en la escritura combinada de dos dígitos: el cero y el uno. En consecuencia, la forma de guardar información en cinta, es mandar dos señales diferentes, una para cada dígito. Y la manera de entender los datos que recibe desde el grabador es distinguir cuándo llega un cero y cuando un uno. Para esto se adoptó darle a cada dígito una señal estandar conocida con el nombre de CUTS. La señal de 8 ciclos de 2.400 Hz identifica al 1 y 4 ciclos de 1.200 Hz es para identificar al cero.

Esta frecuencia se adquirió por la mayoría de las computadoras personales por ser las señales que la mayoría de ellos pueden ejecutar

Otra manera de identificar cada dígito es que al valor 1 se le de un sonido largo y al 0 otro bastante más corto. Por eso, la computadora solo tendrá que diferenciar los tiempos que duran cada señal y así asignarle el valor correspondiente a la señal enviada.

Cuando el polo norte de las partículas de hierro apuntan hacia la derecha, representan al cero. En cambio si apuntan a la izquierda quiere decir que se trata de un uno.

Esta forma de diseñar diferentes campos magnéticos se denomina "saturación" porque no existen posiciones intermedias de los polos de las partículas. Es decir que siempre apuntan hacia una dirección o hacia la contraria.

Al usar cintas comunes, las mismas que utilizamos para guardar música, los datos y programas son almacenados a continuación uno detrás del otro, hasta el final de la cinta.

Esto deja, lamentablemente, sectores sin

FIGURA 1

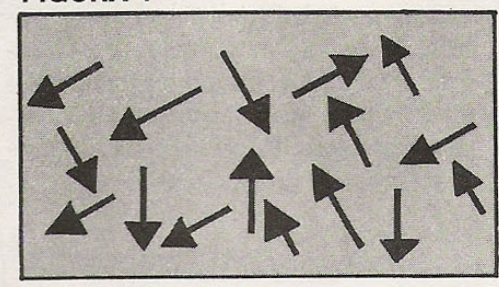
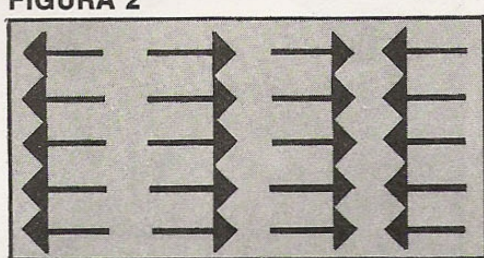


FIGURA 2



poder aprovechar pero que son necesarios para identificar el comienzo y final de la información.

Antes de comenzar a alinear los componentes de la cinta, hace falta crear una pequeña magnetización. Las señales recibidas son amplificadas hasta 4.000 Hz durante la grabación mientras que al reproducir es disminuida hasta 400 Hz.

Pero para borrar las antiguas señales que existían en las cintas, la mayoría de los grabadores utilizan un cabezal y una corriente DC para borrar, o bien una señal ultrasónica.

Si las cintas se exponen entre campos magnéticos alternativos, los polos de las partículas de hierro cambian de orientación, en cambio si se aleja paulatinamente de los campos magnéticos, la alineación de las partículas es casi aleatoria y se produce así un campo magnético nulo.

No es necesario el mas caro de los grabadores.

Es común pensar que cuanto más sofisticado sea el aparato de grabación, mejores serán los resultados del paso de datos.

Pero ésta es una suposición errónea porque los artefactos tan sofisticados poseen una serie de filtros y circuitos

que tienden a reformar la señal hasta convertirla en una débil que, en la mayoría de las computadoras, produce error de entrada/salida de información. Esto se debe a que los completos componentes de audio están fabricados para repoducir o grabar efectos sonoros con la mejor fidelidad y lo más agradable a nuestros oídos como sea posible. Y menos aun hace falta que el grabador sea estéreo. Si es mono la mayoría de los cassettes grabados, que adquiramos en los comercios, podrán ser cargados en las computadoras.

Si disponemos de un grabador estéreo utilicemos solo alguno de los dos canales para realizar nuestras grabaciones. Este tipo de equipos de audio graban en dos pistas diferentes, y una mala alineación de la cabeza del grabador, provocará una señal mono muy baja y en consecuencia poco descifrable por nuestra máquina.

También sería aconsejable no utilizar los los reproductores de sonido con control de tono automático, pero si lo tiene, habrá que aumentar los agudos al máximo. Lo mismo sucede con el control de volúmen porque éstos suelen cambiar el volúmen justo en zonas claves provocando confusión en los datos, cambiando el significado de los mismos y creando consecuentemente un error. En cambio, manualmente nosotros controlamos el volúmen adecuado para la computadora. Por supuesto después de intentar varias veces grabar y reproducir datos desde una máquina hacia la otra.

Por eso aconsejamos grabadores sencillos que solamente funcionen correctamente.

Problemas de grabación

Tal vez el grabador que usabamos siempre repentinamente se encapriche impidiéndonos pasar la información grabador—computadora. Por supuesto pasará por nuestra mente la incertidumbre debido a que nunca antes habíamos tenido algún problema e inesperadamente nos es imposible continuar con la transmisión de datos.

Este problema puede deberse a la obstrucción del cabezal por partículas magnéticas, suciedad y polvillo que es inevitable encontrar en el ambiente. Al obstruir el cabezal, el campo magnético que este produce queda atrapado en un circuito completo. Esto es provocado, frecuentemente, también por las materias magnetizadas que componen las cintas. Como consecuencia, obtendremos una señal débil de reproducción.

Pero se puede corregir fácilmente limpiando el cabezal con papel tisu humedecido con alcohol.

Pero no es el único inconveniente que podemos encontrar, también es común que se produzca una desviación del cabezal, cambiando el ángulo de lectura—escritura.

La forma de notar este síntoma es generalmente no poder cargar los programas grabados por otros artefactos de audio.

La ranura del cabezal mantiene un ángulo con respecto al borde la cinta que normalmente suele ser de 90 grados. Al pasar la cinta por la ranura, como explicamos antes, se magnetiza por sectores, y al cambiar el ángulo, se pueden llegar a leer porciones de los sectores provocando cambios al verdadero significado de la información.

La figura 4 esquematiza la alineación entre el cabezal y la cinta.

La desalineación del cabezal implica poder reproducir solamente los datos que fueron grabados con el mismo ángulo que tiene en ese momento la cabeza lectora.

La forma de corregir este inconveniente también es bastante sencillo y consiste en introducir un pequeño destornillador y mover el tornillo ubicado cerca del cabezal.

Es conveniente, para saber cuando realmente está ajustado el ángulo, tratar de reproducir un cassette con un programa moviendo dicho tornillo hasta lograr que el sonido sea lo más nítido y fuerte como sea posible.

Hace falta tener cuidado con el trato del pequeño tornillo porque si lo giramos demasiado podríamos romperlo, agregando aun más problemas.

Otro caso posible es encontrarnos con el desgaste del cabezal pues están fabricados (la mayoría) por material relativamente blando que permite un desgaste paulatino del mismo. Aquí no hay otra solución que comprar una cabeza grabadora—lectora nueva.

Tratemos de no exponer las cintas cerca de transformadores y artefactos eléctricos como televisores ni exponerlas a los rayos solares, porque como explicamos arriba, estas son modificadas por los imanes.

Pero a pesar de los muchos inconvenientes que podemos encontrar con este sistema de archivar información, sigue siendo el método mas económico y utilizando, aunque tambien el más lento.

FIGURA 3

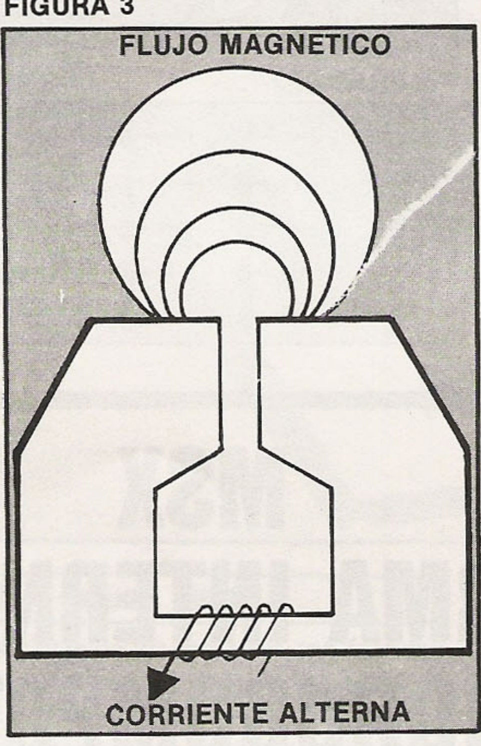
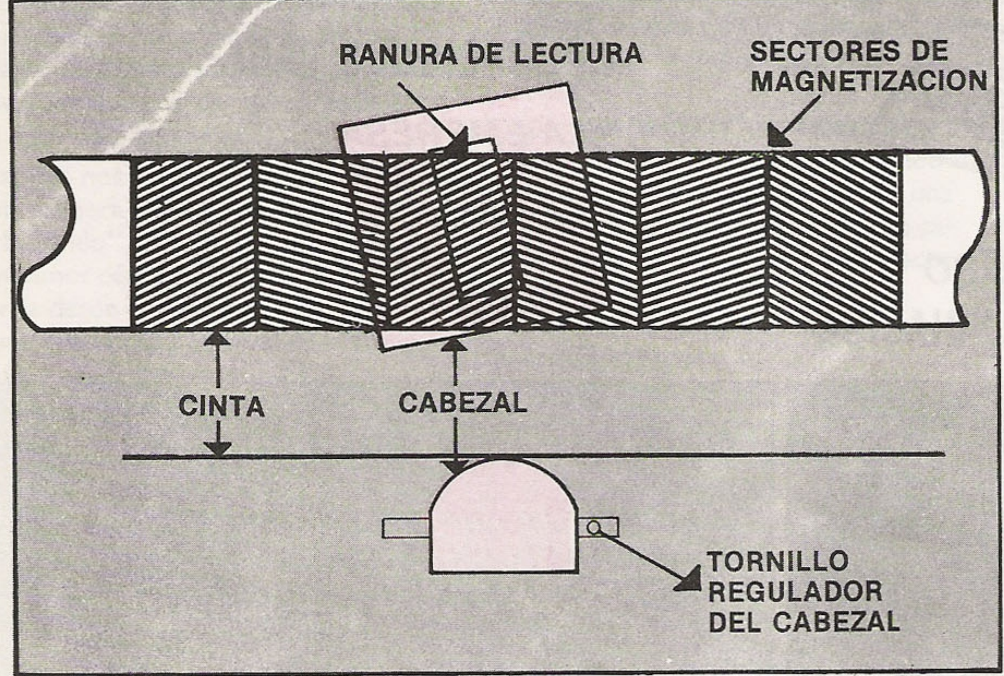
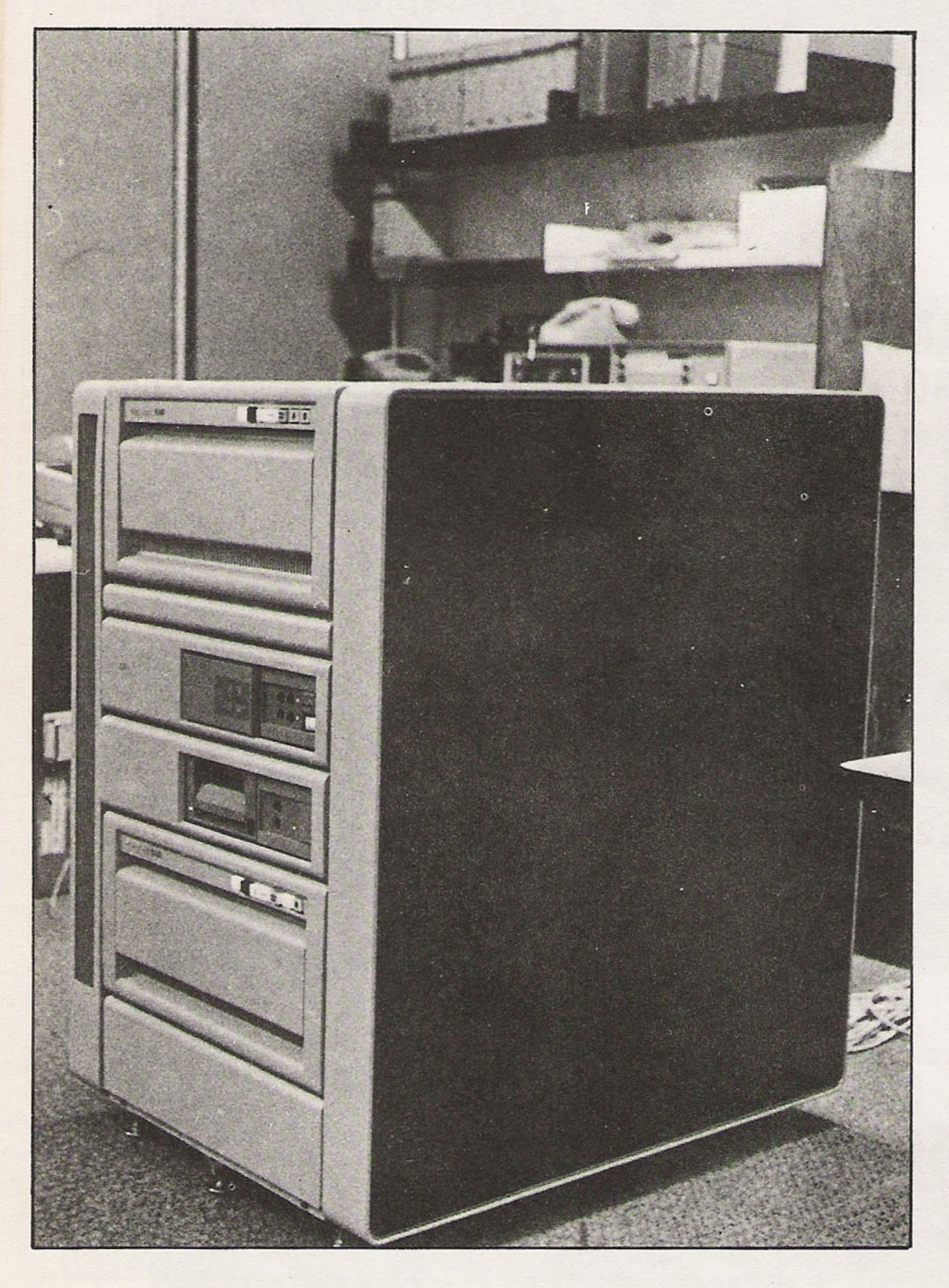


FIGURA 4



BASES DE DATOS PARA UN GRAN MODEM

En numeros anteriores hemos presentado formalmente al periférico que hace posible la comunicación informática o telemática. Veremos ahora qué nos ofrece nuestro país y sus empresas al respecto.



Unidad Central de Proceso de Siscotel.

Tres empresas han desarrollado y llevado a cabo sus ambiciosos proyectos, sobre bases de datos. El primero, fruto de un acuerdo entre Telemática (Talent) y el Banco del Buen Ayre, permite a los clientes de esta entidad mantenerse informados sobre el estado de sus respectivas cuentas.

El enlace que se realiza por medio de una consola Talent DPC-200 y su respectivo modem telefónico, posibilita al usuario obtener datos del tipo de los que ofrece un cajero automático corriente, entre ellos: saldos, transferencias, movimientos del día anterior, detalle de inversiones a vencer, detalle de facturas a vencer, resúmen de otros meses, cambio de claves, movimientos del día, movimientos a debitar o acreditar, detalle de compras a debitar, resúmen del mes anterior, pedido de chequeras y boletas, y correo electrónico.

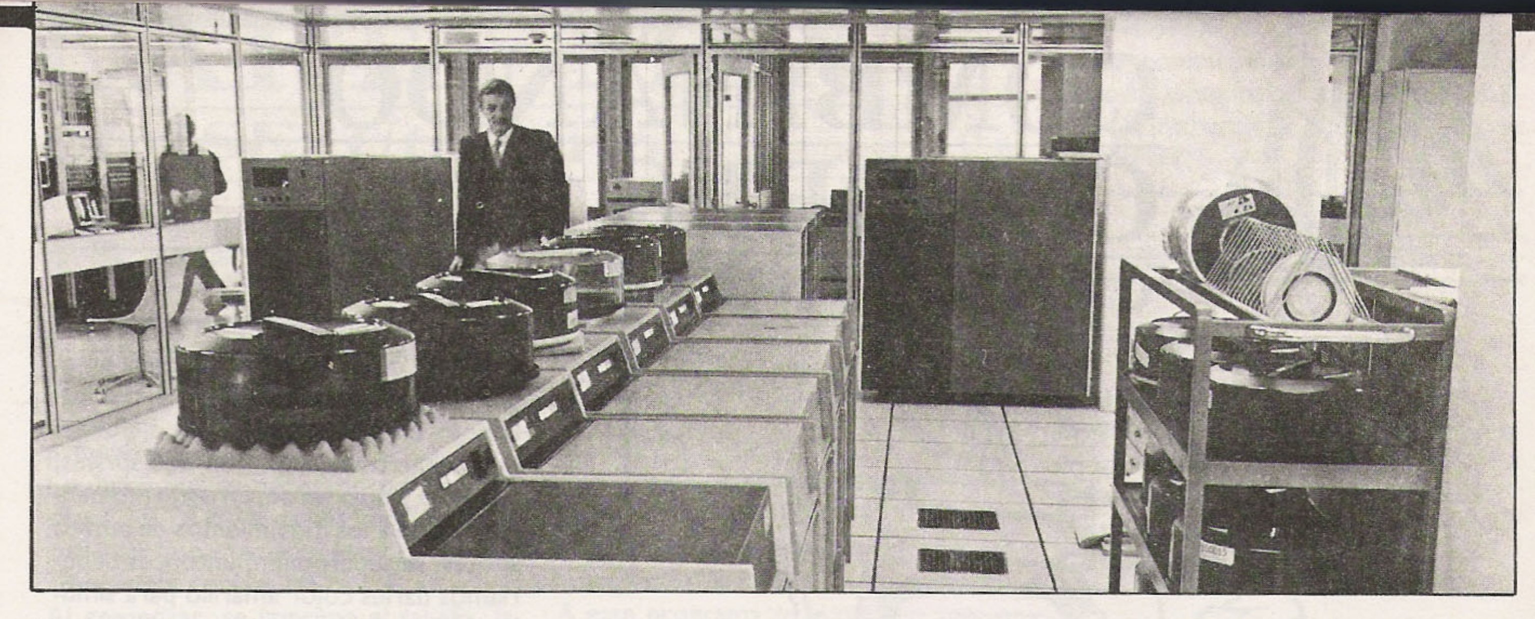
Como vemos las posibilidades que abarca este desarrollo son, para el usuario, muchas y variadas. Por supuesto todo esto es posible gracias a un programa hecho por el mismo banco, permitiéndole al cliente hacer su propia conciliación bancaria, por medio de la opción de mantener detalles en papel por impresora. Este programa, almacenado en parte de la ROM del modem, se ejecuta automáticamente al ser llamado por medio de un Call a dicha ROM.

El sistema hard completo para acceder a este sistema es vendido también como parte del servicios que ofrece a la mencionada entidad. Recordemos que dentro de los ochenta kilobytes de ROM del modem telefónico fabricado por Talent, se encuentran los programas necesarios para acceder a este banco de datos electrónico, un procesador de textos y una planilla electrónica de cálculo.

DELPHI

Otra de las opciones la presenta SISCOTEL, empresa argentina que posibilita el acercamiento a las más grandes bases de datos nacionales e internacionales. A través de ella, el usuario está conectado con una de las más vasta empresa de información mundial: DELPHI.

Delphi no solo es una monstruosa base



Alta tecnología en el Banco del Buen Ayre.

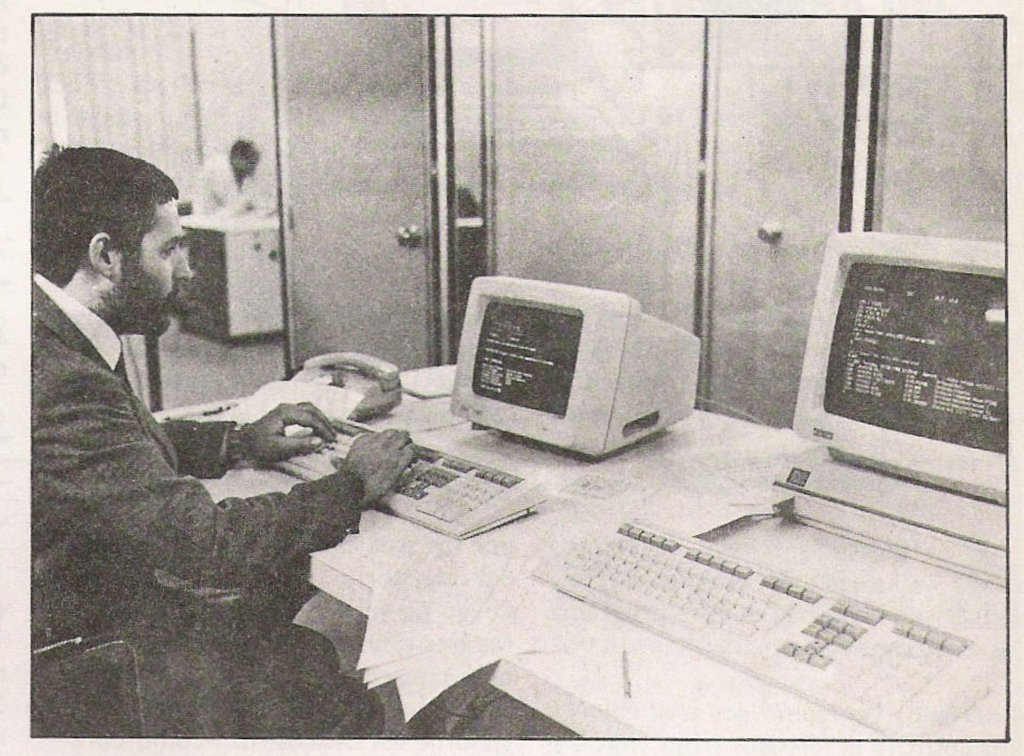
de datos estadounidenses sino que además está dotada de miles de tentáculos que conectan a dicha base con los centros de información de todo Norteamérica y del mundo, y por ende al mismo usuario.

La lista de posibilidades, al igual que en todos estas grandes empresas de información electrónica, es numerosa. Entre ellas encontramos: información periodística, entretenimientos on line, información financiera, telex, correo electrónico nacional y extranjero entre usuarios, impresión de cartas del usuario ccon su correspondiente membrete y logotipo y posterior envio a correo común, compras desde la computadora, horarios de salidas de aviones, trenes y omnibus, biblioteca electrónica, acceso a información técnica y profesional de cualquier área, etc.

Lo que más llama la atención al entrar a las oficinas de Siscotel es lo moderno, ordenado, silencioso y poco computarizado de sus salas.

Hasta donde llegamos, un gran sistema UT 220 de DIGITAL, silencioso y eficiente, casi no se percibe. Lo mismo ocurre con la memoria y el procesador central, de menor tamaño que un pequeño escritorio, cuya capacidad es de 1 Giga byte (10^6 kbytes aproximadamente), y se encuentra aislado en una habitación de cristal bastante grande para sus necesidades.

Así, entrar en SISCOTEL es entrar al futuro. Desde una elegante oficina en la que la cibernética se traduce en una terminal del UT 220, un modem telefónico, una impresora y por supuesto un teléfono; una pequeña demostración de comunicación con las listas de clubes y revistas que poseen espacio en una de las bases de datos de Delphi en Massachusets (EE.UU.), tarda unos pocos segundos en maravillarnos.



Siscotel nos conecta con Delphi.

Nuestra revista ''Load para usuaria de MSX'' estará prestando un servicio dentro de las páginas electrónicas de DELPHI. Nuestros lectores y los usuarios de esta gran base de datos podrán acceder a información, programas y correo electrónico que nuestra editorial renovará y pondrá a disposición de todos aquellos que poco a poco se vayan sumando a la nueva fantasía electrónica.

Realmente el modem de Telemática nos permite ingresar al universo de la información, que se regenera y expande a mucha mayor velocidad de lo que podemos llegar a explorar. Esto lo hace sencillamente infinito...

The Source

Infotel nos conecta con otro de los más grandes servicios en bases de datos mundiales: The Source, dependiente del Reader's Digest Association.

También proporciona informaciones de todo tipo que son actualizadas cada hora con las noticias más importantes, abarçando las financieras y comerciales. Dispone de servicios para los inversores. Por ejemplo, se puede acceder a cotizaciones provistas por la New York Stock Exchange y American Stock Exchange, además de asesoramiento de inversiones y de informes sobre más de 300 industrias y compañías.

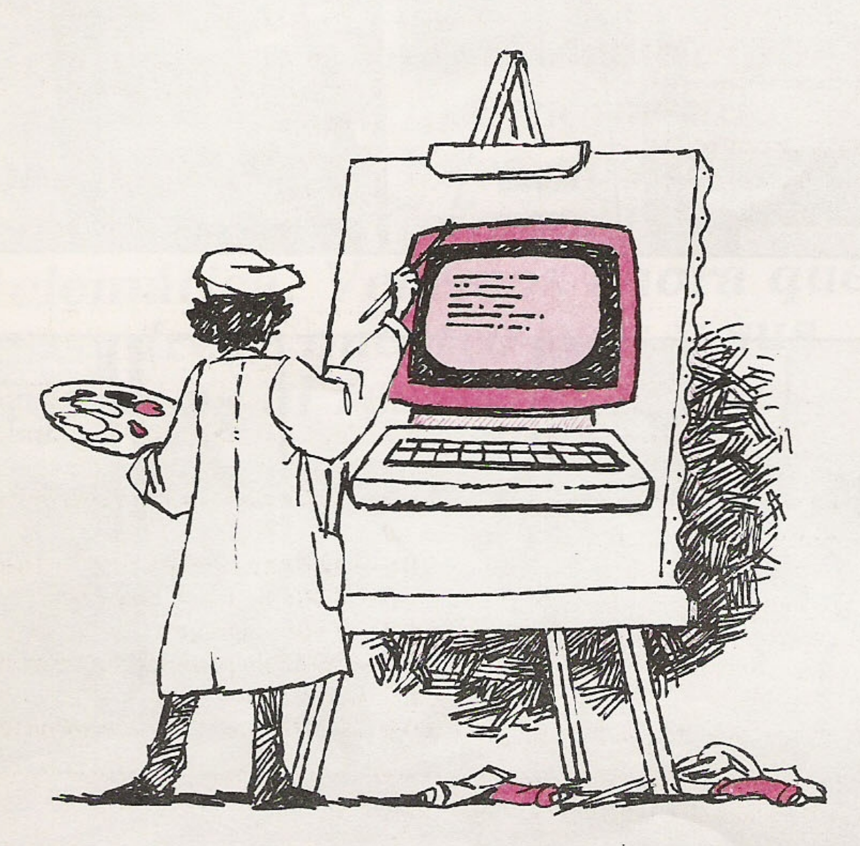
Permite el intercambio de correspondencia electrónica y también ofrece una guía para el comprador de software y de hardware, y otra guía de informaciones turísticas y de vuelos.

En relación a educación permite el acceso a la Groiler's Academic American Encyclopedia que ofrece miles de artículos de un amplio espectro de temas, que van desde arte e historia hasta biografías y tecnología.



COMBINANDO COLORES

CLASE: Utilitario



Las combinaciones de colores pueden tener distintos efectos, y la utilidad de este programa es la de mostrarlos.

Podremos ver cómo los tres colores básicos para el televisor (rojo, verde y azul), forman diferentes gamas.
Al hablar de los colores primarios se refiere a los básicos que son disparados por los tres rayos ca tódicos del tubo del televisor.
Luego de ver las combinaciones de estos tres tonos, se obtienen como resultado tres colores (amarillo, magenta y cyan), que al complementarlos con alguno de los básicos, dan como consecuencia el blanco.
La aplicación de este utilitario es sumamente sencilla, permitiendo agregarle utilidades que necesitemos, como otras combinaciones de tonos.
Demos un ejemplo práctico; supongamos estar trabajando con dos sprites, uno rojo y otro verde. En algún momento, debido a los movimientos de ambos sprites, se superponen; entonces deberíamos darles color amarillo para simular mejor el efecto de superposición.
Pero la conclusión de pintar a los sprites de amarillo fue el resultado de ver qué pasaba (a través de COMBINANDO COLORES) cuando juntábamos el rojo y verde.

VARIABLES IMPORTANTES;

A$: respuesta para seguir ejecutando el programa

ESTRUCTURA DEL PROGRAMA:

10—110: mensajes al comienzo del programa.
130—240: efectos con colores primarios.
250—320: primer efecto con colores secundarios
330—390: segundo efecto con secundarios
400—470: tercer efecto

```
10 KEY OFF
20 COLOR 11,1,1
30 SCREEN 2
40 OPEN"GRP:"FOR OUTPUT AS#1
50 PSET(10,10),1: PRINT#1,"COLOR
ES PRIMARIOS Y SECUNDARIOS"
60 PSET(10,18),1:PRINT#1,"-------
-------------------------------"
70 PSET(10,40),1:COLOR 9:PRINT#1
,"CUAL QUIERE VER?"
80 PSET(10,100),1:COLOR 5:PRINT#
1,"1. SUMA DE COLORES PRIMARIOS"
90 PSET(10,116),1:COLOR 5:PRINT#
1,"2. SUMA DE COLORES SECUNDARIOS
"
100 PSET(42,144),1:PRINT#1,"ENTR
E SU ELECCION:"
110 A$=INKEY$: IF A$="1" THEN GO
SUB 130 ELSE IF A$="2" THEN GOSUB
 260 ELSE GOTO 110
120 CLS:GOTO 50
130 CLS: LINE (50,20)-(100,90),6
,BF
140 LINE (100,20)-(120,90),10,BF
150 LINE (120,20)-(170,90),12,BF
160 LINE (85,50)-(135,120),4,BF
170 LINE (85,50)-(100,90),13,BF
180 LINE (100,50)-(119,90),15,BF
190 LINE (120,50)-(135,90),7,BF
200 PSET (10,130),1:COLOR 15:PRI
NT#1,"ROJO  + VERDE  = AMARILLO"
210 PSET (10,138),1:PRINT#1,"ROJ
O  + AZUL  = MAGENTA"
220 PSET (10,146),1:PRINT#1,"VER
DE  + AZUL  = CYAN"
230 PSET (10,181),1:PRINT#1,"PRE
SIONA LA BARRA DE ESPACIO"
240 A$=INKEY$: IF A$<>" " THEN 2
40
250 CLS:RETURN
260 CLS:PSET(10,10),1:PRINT#1,"S
UMA DE COLORES SECUNDARIOS"
270 LINE (50,50)-(100,120),6,BF
280 LINE (100,50)-(120,120),15,B
F
290 LINE (120,50)-(170,120),7,BF
300 PSET (10,140),1:COLOR 15:PRI
NT#1,"ROJO  + CYAN  = BLANCO"
310 PSET (10,181),1:PRINT#1,"PRE
SIONA LA BARRA DE ESPACIO"
320 A$=INKEY$: IF A$<>" " THEN 3
20
330 CLS:PSET(10,10),1:PRINT#1,"S
UMA DE COLORES SECUNDARIOS"
340 LINE (50,50)-(100,120),4,BF
350 LINE (100,50)-(120,120),15,B
F
360 LINE (120,50)-(170,120),10,B
F
370 PSET (10,140),1:COLOR 15:PRI
NT#1,"AZUL  + AMARILLO  = BLANCO"
380 PSET (10,181),1:PRINT#1,"PRE
SIONA LA BARRA DE ESPACIO"
390 A$=INKEY$: IF A$<>" " THEN 3
90
400 CLS:PSET(10,10),1:PRINT#1,"S
UMA DE COLORES SECUNDARIOS"
410 LINE (50,50)-(100,120),12,BF
420 LINE (100,50)-(120,120),15,B
F
430 LINE (120,50)-(170,120),13,B
F
440 PSET (10,140),1:COLOR 15:PRI
NT#1,"VERDE  + MAGENTA  = BLANCO"
450 PSET (10,181),1:PRINT#1,"PRE
SIONA LA BARRA DE ESPACIO"
460 A$=INKEY$: IF A$<>" " THEN 4
60
470 CLS:RETURN
```

CARTA MSX

CLASE: Utilitario

Con este programa podremos escribir e imprimir cartas y sobres a través de nuestra computadora.

Su utilización es sencilla y solamente basta seguir las instrucciones que se nos va indicando durante el transcurso del programa.

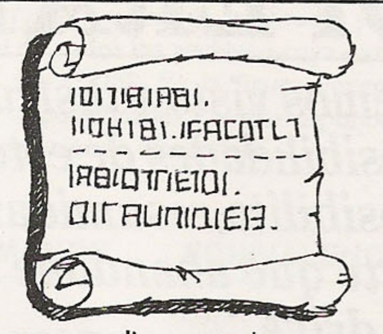
Lo único que es necesario para poder usar CARTA MSX es una impresora de cualquier tipo y tamaño, siempre que sea compatible con las MSX.

Al comenzar, se imprime el saludo, la inicialización de la carta y luego línea por línea de la misma.

Esto significa que al concluir un renglón se deberá presionar ENTER y seguir escribiendo. El efecto que puede provocar escribir todas las líneas de golpe, es que serán impresas una al lado de la otra según sea el ancho de la impresora.

Si sabemos manejar la impresora, formateándola para el ancho que preferamos, podremos borrar las líneas 340—350—360 y 370.

Como esto depende de cada impresora, dejamos esta opción para que lo



piensen y realicen ustedes.

Al terminar el cuerpo de la carta, imprimiremos el saludo.

A este programa se le pueden agregar las opciones que cada uno necesite, pero básicamente se presta para cualquier utilidad.

Como escribirle a un amigo, al jefe pidiéndole aumento o bien mandándonos algún mensaje a nosotros.

Pero por ejemplo, se le podría agregar que imprima nuestro nombre en el ángulo inferior derecho de la carta, el código postal en el sobre, etc.

Seguramente a medida que sea utilizado se darán cuenta de las variantes realizables a este utilitario.

Cuando se nos pide entrar las respuestas a las preguntas que nos va formulando la computadora, no aparecerá el signo de pregunta porque hemos utilizado la instrucción LINE INPUT en vez de INPUT.

VARIABLES IMPORTANTES:

S$: saludo
L$: línea de la carta
D$: despedida
OM$ nombre del destinatario de la correspondencia
D$: dirección
P$: localidad
C$: provincia

ESTRUCTURA DEL PROGRAMA

10—170: mensaje con las instrucciones
180—290: para escribir/imprimir el saludo
300—450: escribe/imprime las líneas
460—580: imprime despedida
590—710: mensaje
720—890: acepta/imprime datos del destinatario
900—920: mensajes de finalización

```
10 SCREEN 0:COLOR 11,1,1:KEY OFF
20 PRINT"««««««««««««««««««««««««
««««««««««««««"
30 PRINT"«
        «"
40 PRINT"«          *** - C A R T A
 - ***        «"
50  PRINT"«
         «"
60 PRINT"«          *** -   M S X
 - ***        «"
70 PRINT"«
        «"
80 PRINT"««««««««««««««««««««««««
««««««««««««««"
90 PRINT,,"Primero se entra la i
ntroducción del saludo como por e
jemplo 'estimado    cliente/ amig
o'"
100 PRINT"Cuando hayas escrito e
l saludo          inicial, presicna
« ENTER » y se      imprimirá."
110 PRINT "Deberás entrar línea
por línea para  imprimirla y cuan
do quieras terrminarentra « N »."
120 PRINT"No utilices comas"
130 PRINT"Y recuerda presionar «
 ENTER » al      terminar cada line
a"
140 PRINT "Enciende la impresora
 y luego pre-    siona « ENTER »."
150 LINE INPUT A$
160 IF A$="" THEN 180
170 END
180 CLS
190 PRINT"** E S C R I B I R   S
 A L U D O **"
200 PRINT ,,"SALUDO:";: LINE INP
UT S$
210 LOCATE 2,10: PRINT"ENCIENDE
LA IMPRESORA"
220 PRINT,,"PRESIONA « ENTER »";
230 LINE INPUT E$
240 IF E$="" THEN 260
250 END
260 CLS
270 LOCATE 2,16: PRINT"IMPRIMIEN
DO"
280 LPRINT TAB(20);S$
290 LPRINT TAB(3)" "
300 CLS
310 PRINT " ESCRIBE LAS LINEAS"
320 PRINT "--------------------"
330 PRINT,,"LINEA: ";:LINE INPUT
 L$
340 PRINT,,"PRESIONA « ENTER »";
350 LINE INPUT U$
360 IF U$="" THEN 380
370 END
380 CLS
390 LOCATE 2,16: PRINT"IMPRIMIDO
"
400 LPRINT TAB(4);L$
410 LPRINT TAB(3)" "
420 PRINT"    PARA INTERRUMPIR LA
 IMPRESION "
430 PRINT:PRINT"       PRESIONAR
 « N »";:LINE INPUT N$
440 IF N$="N" OR N$="n" THEN 460
450 CLS:GOTO 310
460 CLS
470 PRINT " D E S P E D I D A "
480 PRINT "-------------------"
490 PRINT, "DESPEDIDA: ";:LINE I
NPUT D$
500 LOCATE 2,15: PRINT"IMPRIMIEN
DO"
510 PRINT,,"PRESIONE « ENTER »";
520 LINE INPUT P$
530 IF P$="" THEN 550
540 END
550 CLS
560 LOCATE 2,15: PRINT"IMPRIMIEN
DO"
570 LPRINT TAB(4)" "
580 LPRINT TAB(21);D$
590 CLS
600 LOCATE 2,15: PRINT"FIN DE LA
 CARTA"
610 PRINT,,"PRESIONE « ENTER »";
620 LINE INPUT PU$
630 IF P$="" THEN 650
640 END
650 CLS
660 PRINT,,"COLOCA EL SOBER A LA
 DERECHA DE LA IMPRESORA."
670 PRINT,,"UNA VEZ COLOCADA RES
PONDE LAS PRE-   GUNTAS DEL OREDE
NADOR."
680 LOCATE 2,15:PRINT "PRESIONA
« ENTER »";
690 LINE INPUT EN$
700 IF EN$="" THEN 720
710 END
720 CLS
730 PRINT " D A T O S   D E L
C L I E N T E"
740 PRINT "-----------------------
---------------"
750 PRINT "NOMBRE: ";:LINE INPUT
 OM$
760 PRINT,, "DIRECCION: ";:LINE
INPUT D$
770 PRINT,, "LOCALIDAD: ";:LINE
INPUT P$
780 PRINT,, "PROVINCIA: ";:LINE
INPUT C$
790 LOCATE 2,14: PRINT"IMPRIMIEN
DO:"
800 PRINT,, "PRESIONA « ENTER »"
;
810 LINE INPUT SA$
820 IF SA$="" THEN 840
830 END
840 CLS
850 LOCATE 2,15: PRINT"IMPRIMIEN
DO"
860 LPRINT TAB(8);OM$
870 LPRINT TAB(7);D$
880 LPRINT TAB(7);P$
890 LPRINT TAB(20);C$
900 CLS
910 LOCATE 2,15: PRINT"FIN DE CA
RTA"
920 PRINT "QUIERES ESCRIBIR OTRA
? (S/N)":LINE INPUT R$:IF R$="S"
OR R$="s" THEN RUN ELSE END
```

EL SISTEMA (4ta. parte)
OPERATIVO DOS

Hemos visto y disfrutado ya de muchas de las posibilidades de este grandioso software, que posibilita,resumidamente, el manejo de todo byte que abandone a la consola MSX rumbo al drive

Continuamos viendo las posibilidades que nos ofrece el comando COPY.

Como habíamos visto, éste cumple la función de copiar de un periférico cualquiera a otro, cualquier tipo de archivo (programas Basic y Assembler, listados de números o letras, pantallas, anything).

Si por ejemplo trabajamos dentro del mismo drive de disquettes, una de las opciones puede tomar la siguiente forma:

COPY JULIO. DAT CESAR.DAT

Donde el archivo JULIO.DAT será grabado dentro del mismo drive con el nombre CESAR.DAT. Otra posibilidad es la siguiente:

COPI HILDA.DAT+MARTIN.DAT MATIAS.DAT que creará su archivo llamado MATIAS.DAT con la concatenación de los archivos HILDA.DAT y MARTIN.DAT. Notemos que no hace falta ningún tipo de símbolo adicional para efectuar la copia, entre los nombres de los archivos.

Teniendo ya dos drives de discos, las mismas copias pueden efectuarse de un disco contenido en un drive a otro. Por ejemplo: COPY A:BELTRAN.DAT+B:ROCA.DAT NEGRO.RIO

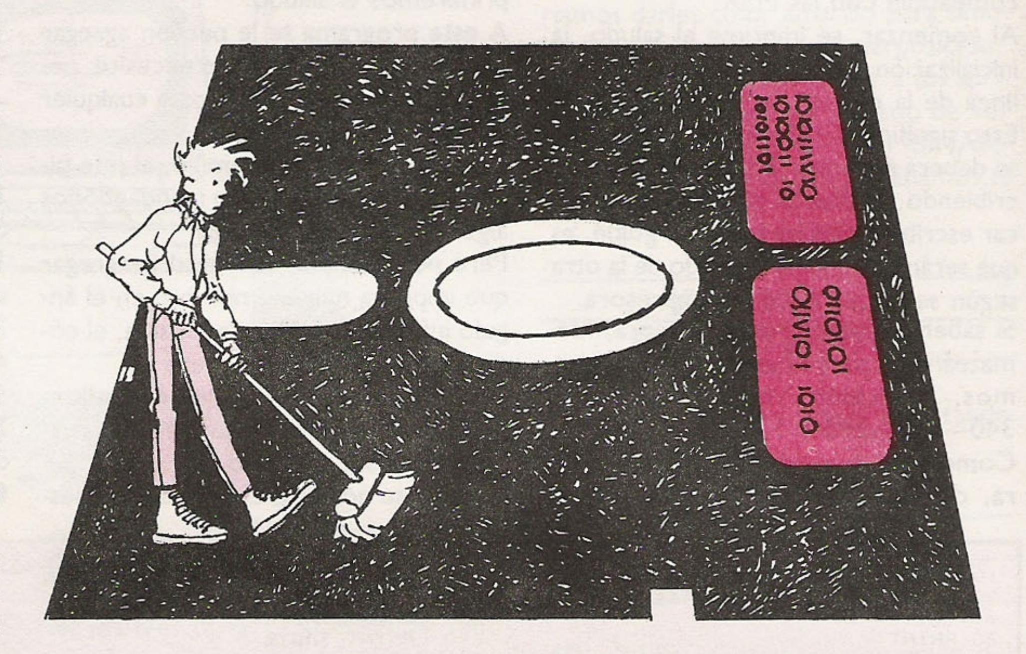
Este comando tomará el archivo BELTRAN.DAT del drive llamado A, el archivo ROCA.DAT del drive B, los concatenará y los grabará bajo el nombre NEGRO.RIO en el drive en que se estaba trabajando antes de llamar a este comando. Así es que se llama a este drive de defecto.

Otra de las muchas opciones es la de unir todos los archivos que posean la misma extensión y pasarlos a un nuevo archivo en algunos de los periféricos, ejemplo;

COPY *.DAT TODOS.DAT

Juntará todos los archivos cuya extención o terminación sea DAT y los grabará bajo el nombre TODOS.DAT en el mismo disco.

Generalmente los archivos que se suelen encadenar son los del tipo ASCII, que son aquellos que tienen algún significado del tipo de los textos. Es asi como este comando necesita de un marcador de fin de archivo, y utiliza para esto el caracter CTRL+Z cuyo valor ASCII es lA en hexadecimal. Entonces para realizar una concatenación es necesario que tenga un solo caracter de fin de archivo al final de la suma de ellos. Para ello se utiliza "/B" que le informa al comando COPY que debe leer hasta el final



físico del archivo, antes de juntarlo al siguiente.

El final físico de un archivo es el que corresponde a su longitud total, que es aquella que figura junto al nombre del archivo cuando ejecutamos el comando DIR.

Si a modo de ejemplo ejecutamos el comando:

COPY/B X.DAT+Y. DAT

se efectuará la copia del archvo Y.DAT a continuación de X.DAT con el nombre X.DAT.

Cualquier tipo de archivos pueden ser unidos por medio de "/B" para los que sean de tipo binario y con "/A" para los de formato ASCII.

Repasemos. Usando /B el archivo es leído y copiado en toda su extensión, con todos los CTRL—Z que contenga y usando /A se eliminarán los CTRL—Z dejando solamente uno al final del archivo.

Con este comando, como dijimos, podemos efectuar copias entre distintos periféricos. Para efectuar este tipo de trabajo con un dipositivo externo que no sea la disquettera, debe especificarse cual de ellos es.

Los nombres que representan a los distintos tipos de dispositivos son los siguientes:

AUX: se lo usa para hacer referencia a una entrada o salida de datos desde o hacia un periférico auxiliar (impresora o drive).

CON: indica que el traspaso de información se hará a través de la consola. Si es de entrada, se asumirá el teclado y si es de salida el monitor o televisor.

LST o PRT: hace referencia directa a la impresora.

Veamos algunos ejemplos:

COPY MAINQUE.DAT LST copiará el archivo llamado MAINQUE.DAT en la impresora, efectuando en ella la representación ASCII de cada uno de los bytes que componga a dicho archivo, que por supuesto será buscado en la disquettera.

COPY MAINQUE.DAT CON hará también una copia del mismo archivo pero esta vez sera visualizado en la pantalla.

COPY CON PRT hará una copia en la

impresora de todo lo que tipeamos en el teclado luego de entrar el comando y hasta que se llene el buffer o entremos el caracter CRTL—Z.

COPY CON CON nos permitirá, al igual que el anterior y de la misma forma, visualizar las teclas que hayamos pulsado luego de entrar dicho comando. También es posible crear un archivo en disco directamente entrando los caracteres desde el teclado. Para ello debemos especificar que usaremos como periférico de partido la consola, y de destino el nombre de un archivo.

Asi, debe tener la siguiente forma: COPY CON PROPERZI. DOC
Por medio del uso adecuado de ciertos nombres de archivos que son fijados por este sistema operativo y de este comando, podremos crear archivos o programas que se autoejecutarán al encender el sistema.
Este tema lo veremos más adelante. Continuemos viendo los restantes comandos.

DATE:

sirve para cambiar la fecha que se ha entrado u obviado al iniciar el sistema. Su uso es muy simple y no merece mayor explicación.

DEL:

puede también escribirse como DELETE o ERASE, y se la utiliza (como astutamente ya intuímos) para eliminar, aniquilar o hacer desaparecer del disco a un archivo determinado. Para ello deberemos escribir el nombre del archivo en desgracia a continuación de alguna de las tres formas que adopta este comando.
Si deseamos eliminar todos los archivos que se encuentren en un determinado disco, simplemente deberemos efectuar la siguiente entrada: DEL *.*. Al hacercerlo la computadora nos preguntará: Are you sure? (¿esta usted seguro?), y si lo estamos deberemos pulsar la tecla correspondiente a la letra "Y" o simplemente RETURN. De otra forma asumirá que nos hemos acobardado y no lo borrará.

DIR:

seguramente ya nos hemos cansado de usar este comando pero igualmente veremos que opciones nos ofrece y sus equivalencias en lo que a sintaxis se refiere.
En primer lugar recordemos que el asterisco (*) es usado como comodín en lugar del nombre o extensión de un programa o archivo, para englobar a todos aquellos que reunan ciertas características. Por ejemplo DIR *.COM nos mostrará los nombres, longitudes y cantidad de todos los archivos cuya extensión sea COM. En la figura veremos algunas posibilidades y su correspondiente equivalencia sintáctica.

COMANDO	EQUIVALENCIA
DIR	DIR *.*
DIR PROEDI	DIR PROEDI.*
DIR .ASS	DIR *.ASS
DIR .	DIR*

Si al final del comando le agregamos /P nos mostrará el listado que pidamos haciendo una pausa por cada cambio de página.
También podemos incluir la opción /W que hará visible solamente los nombres de los archivos, tratando de entrar la cantidad máxima de nombres por línea.

FORMAT:

funciona en forma idéntica al CALL FORMAT por lo que no le dedicaremos ni una línea más.

MODE:

sirve para cambiar el ancho de pantalla, y debe ser acompañado del número que queremos que represente, en síntesis, hace lo mismo que la instrucción WIDTH de Basic y su utilización es idéntica, así que no gastaremos más tinta en ella.

PAUSE:

dentro de los archivos especiales que se ejecutaban al iniciar el sistema, se encuentran unos llamados del tipo batch. Estos son una especie de programa que se logran en base a algunos de los comandos que estamos viendo, y desde el mismo sistema operativo.
Asi es como PAUSE seguida de un comentario o no, se lo utiliza para efectuar una pausa dentro de esos programas. Generalmente se usan para invitarnos a cambiar el disco o bien para hacernos recordar alguna cosa. Para continuar con la ejecución de este programa batch, simplemente deberemos pulsar alguna tecla, salvo CRTL—C que causará la interrupción del mismo, preguntándonos si realmente queremos inerrumpirlo por medio de: TERMINATE BATCH FILE (Y/N)? a lo que responderemos con Y si es que era esa nuestra intención.

REM:

se usa para mostrar mensajes solamente dentro de los archivos batch. Los únicos separadores válidos que pueden incluirse en el mensaje son TAB y comas.

REN:

sirve para cambiarle el nombre a un archivo ya existente por uno nuevo.
Por ejemplo: REN *.DAT *.DOC hará que todos los archivos que tenían la extensión. DAT ahora tengan la. DOC.

TIME:

este comando nos permitiría, si la DPC—200 tuviese reloj, cambiar y visualizar la hora. De todos modos el comando está preparado en el sistema operativo para aquellas máquinas que si lo tengan.

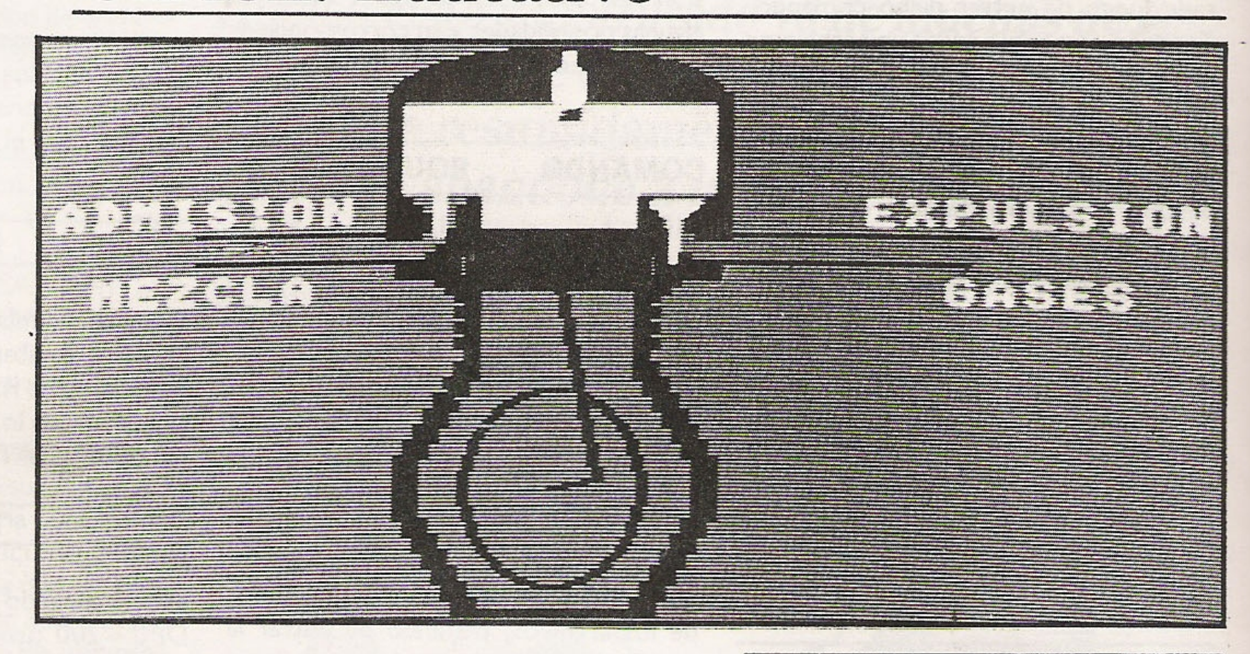
TYPE:

generalmente se lo utiliza para verificar si alguno de los archivos existente en el disco es interpretable como un texto. Lo que realiza en realidad es la interpretación de cada uno de los bytes que componen a un archivo y los muestra en pantalla.
Es similar a una de las opciones que nos proporcionaba el comando COPY (COPY NN CON). Hay que tener en cuenta que si el archivo que intentamos visualizar es del tipo binario (programas o archivos numéricos) serán interpretados como tales los line feed, limpiado de pantalla, beeps o CTRL—Z.
Ejemplo: TYPE FLOWERS.DIR nos dará una representación visual de este archivo.
Y aquí se terminaron los comandos de que disponemos trabajando directamente desde el sistema operativo.
En una próxima entrega veremos algunas aplicaciones interesantes que surgen como producto de la combinación de algunos de estos comandos, en lo que a archivos de autoejecución se refiere.
Por lo demás, podemos ya sacudir el almohadón y volver a sentarnos frente al kiosco de revistas, a esperar otro apasionante número de LOAD MSX.



FUNCIONAMIENTO DEL MOTOR

CLASE: Educativo





Con este programa podremos entender un poco mejor cómo funcionan los motores de nafta.

El gráfico muestra la manera en que el combustible pasa por el múltiple de admisión y llega hasta la válvula que se abre cuando se desalojan los gases de la combustión. La cámara de combustion se llena y el pistón baja movido por la biela (que está sujetada al cigueñal giratorio).

Una vez llena la cámara, se cierra la válvula de admisión, y sube el pistón comprimiendo al máximo la nafta. En ese momento la bujía produce un chispazo provocando la combustión. La válvula de escape, que permite la salida de los gases, se abre mientras baja nuevamente el pistón. El proceso se reitera sucesivamente.

En la pantalla, la nafta tiene color verde y los gases de la combustión rojo.
Con las teclas de cursor hacia arriba se acelera el funcionamiento del motor mientras que el cursor hacia abajo lo hace más lento.
Al mantener la barra de espacio presionada, detendremos el motor.
Si bien no es un programa con gran cantidad de variantes, creemos que su objetivo —explicar el funcionamiento del motor— se cumple por la didáctica que utiliza.
El sonido no es complicado pero es lo que más se acerca a la explosión de un motor.

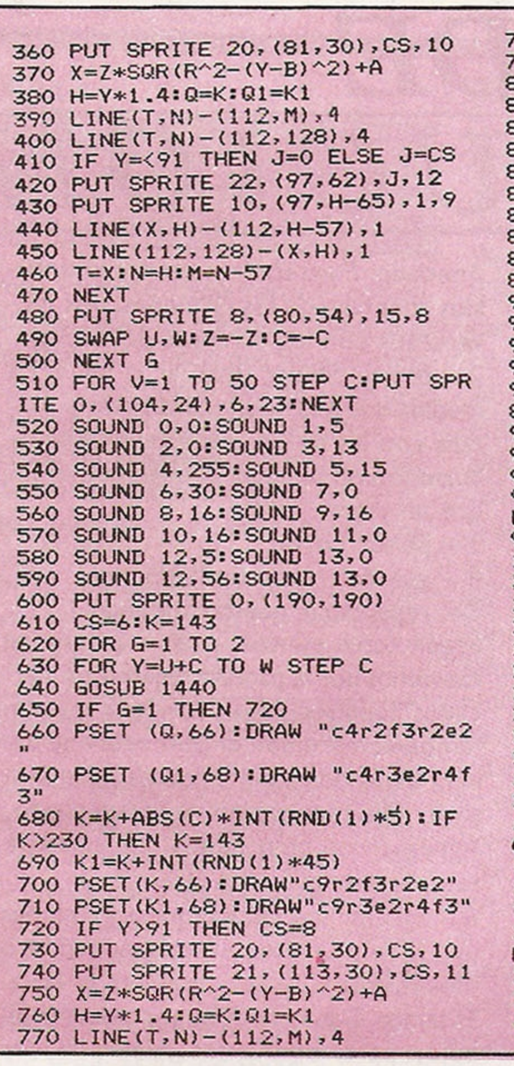
VARIABLES IMPORTANTES:

C: velocidad del motor

ESTRUCTURA DEL PROGRAMA

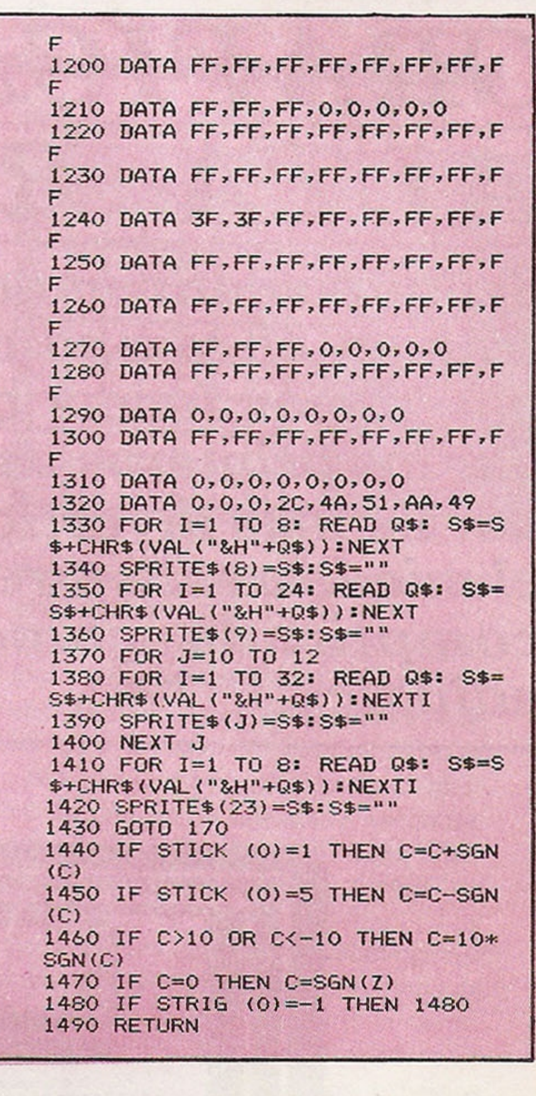
10—150: mensajes por pantalla
170—180: dibuja los múltiples de admisión y escape
190: grafica cigüeñal
200: dibuja bujía
210: ubica la válvula de escape definida por sprite 9
220—230: inicializa variables
240: ubica válvula de admisión
250—500 completa motor y hace la simulación del movimiento
510—600: explosión
610—910: movimiento de expansión y expulsión
920—1140: gráfico del motor
1150:1430: definición de sprites y gra'icos
1440—1490: controla el movimiento del motor (aceleración y retardo).

```
10 SCREEN 0: KEY OFF: COLOR 11,1
,1:CLS
20 LOCATE 7,4: PRINT"MOTOR DE EX
PLOSION"
30 LOCATE 3,9: PRINT"LOS MANDOS
DEL CURSOR "+CHR$(206)+"    "+CHR$
(205)
40 PRINT TAB(3) "REGULAN LA VELO
CIDAD"
50 LOCATE 3,15: PRINT"PARA DETEN
ER EL MOVIMIENTO"
60 PRINT TAB(3)"PULSAR LA BARRA
DE ESPACIO"
70 LOCATE 3,20:PRINT "PARA CONTI
NUAR PULSAR UNA TECLA"
80 IF INKEY$="" THEN 80
90 SCREEN 2,3:COLOR 15,4,7: CLS
100 OPEN"GRP:"AS#1
110 PSET(55,2),4: PRINT #1,"MOTO
R DE EXPLOSION"
120 PSET(10,54),4:PRINT#1,"ADMIS
ION"
130 PSET(18,74),4:PRINT#1,"MEZCL
A"
140 PSET(175,54),4:PRINT#1,"EXPU
LSION"
150 PSET(191,74),4:PRINT#1,"GASE
S"
160 GOTO 920
170 DRAW "c1bm40,63r40bm40,70r40
"
180 DRAW "c1bm144,63r55bm144,70r
50"
190 CIRCLE(112,30),33,1,0,3.14,.
4:PAINT(112,25),1
200 DRAW"C1BM112,34R3U3C15E2U6H2
U3L2D3G2D6F2R2":PAINT(112,28),15
210 PUT SPRITE 9,(127,54),15,8
220 A=112:R=10:B=93:Z=1:C=1
230 U=B-R:W=B+R
240 PUT SPRITE 8,(80,47),15,8:CS
=2
250 FOR G=1 TO 2
260 FOR Y=U+C TO W STEP C
270 GOSUB 1440
280 PSET(Q,65):DRAW"C4R2U1R1F2H2
"
290 PSET(Q1,66):DRAW"C4R2F3R1E3"
300 IF G=2 THEN 340
310 K=K+ABS(C)*RND(1)*3:IF K>80
OR K<40 THEN K=40
320 K1=K1+ABS(2*C)*RND(1)*3:IF K
1>80 OR K1<40 THEN K1=40
330 PSET(K,66):DRAW"c2r2u1r1f2h2
"
340 IF Y<91 THEN CS=3
350 PUT SPRITE 21,(113,30),CS,11
```



```
360 PUT SPRITE 20,(81,30),CS,10
370 X=Z*SQR(R^2-(Y-B)^2)+A
380 H=Y*1.4:Q=K:Q1=K1
390 LINE(T,N)-(112,M),4
400 LINE(T,N)-(112,128),4
410 IF Y=<91 THEN J=0 ELSE J=CS
420 PUT SPRITE 22,(97,62),J,12
430 PUT SPRITE 10,(97,H-65),1,9
440 LINE(X,H)-(112,H-57),1
450 LINE(112,128)-(X,H),1
460 T=X:N=H:M=N-57
470 NEXT
480 PUT SPRITE 8,(80,54),15,8
490 SWAP U,W:Z=-Z:C=-C
500 NEXT G
510 FOR V=1 TO 50 STEP C:PUT SPR
ITE 0,(104,24),6,23:NEXT
520 SOUND 0,0:SOUND 1,5
530 SOUND 2,0:SOUND 3,13
540 SOUND 4,255:SOUND 5,15
550 SOUND 6,30:SOUND 7,0
560 SOUND 8,16:SOUND 9,16
570 SOUND 10,16:SOUND 11,0
580 SOUND 12,5:SOUND 13,0
590 SOUND 12,56:SOUND 13,0
600 PUT SPRITE 0,(190,190)
610 CS=6:K=143
620 FOR G=1 TO 2
630 FOR Y=U+C TO W STEP C
640 GOSUB 1440
650 IF G=1 THEN 720
660 PSET (Q,66):DRAW "c4r2f3r2e2
"
670 PSET (Q1,68):DRAW "c4r3e2r4f
3"
680 K=K+ABS(C)*INT(RND(1)*5):IF
K>230 THEN K=143
690 K1=K+INT(RND(1)*45)
700 PSET(K,66):DRAW"c9r2f3r2e2"
710 PSET(K1,68):DRAW"c9r3e2r4f3"
720 IF Y>91 THEN CS=8
730 PUT SPRITE 20,(81,30),CS,10
740 PUT SPRITE 21,(113,30),CS,11
750 X=Z*SQR(R^2-(Y-B)^2)+A
760 H=Y*1.4:Q=K:Q1=K1
770 LINE(T,N)-(112,M),4
```

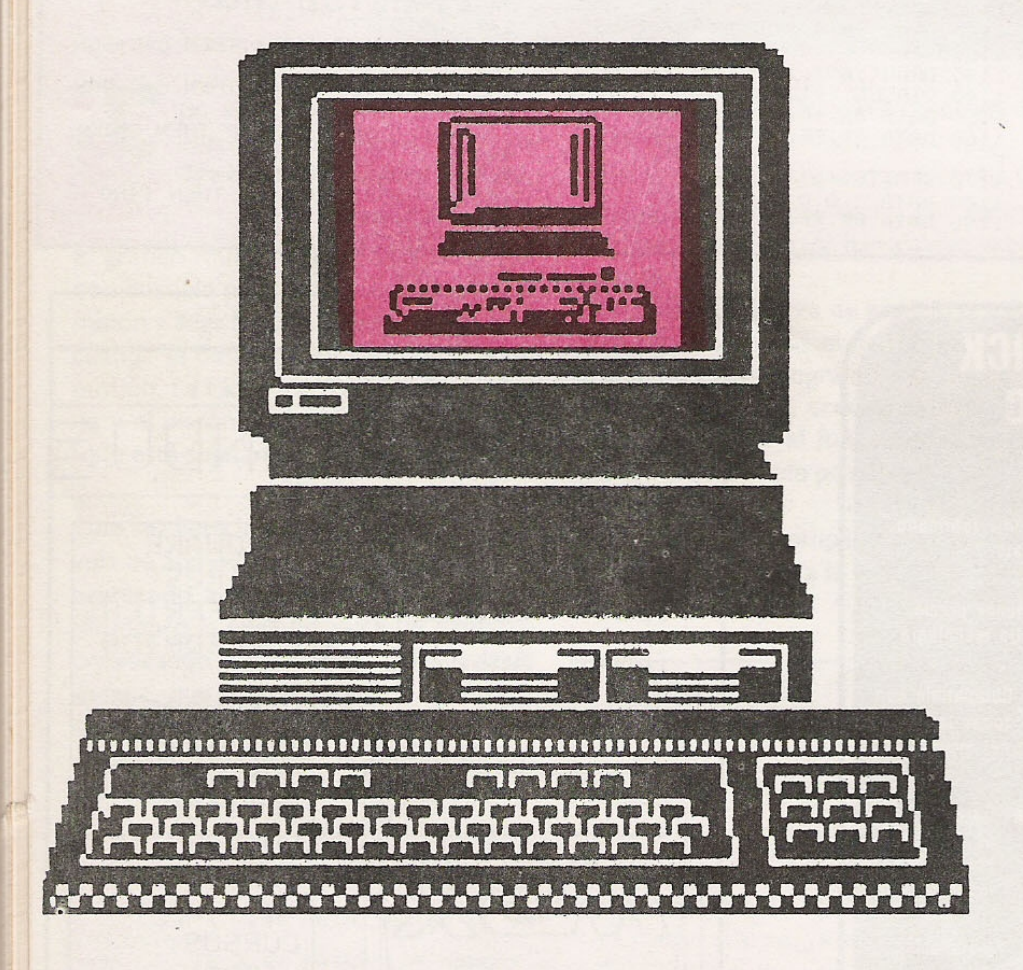
```
780 LINE(T,N)-(112,128),4
790 IF Y=<91 THEN J=0 ELSE J=CS
800 PUT SPRITE 22,(97,62),J,12
810 PUT SPRITE 10,(97,H-65),1,9
820 LINE(X,H)-(112,H-57),1
830 LINE(112,128)-(X,H),1
840 T=X:N=H:M=N-57
850 NEXT
860 PUT SPRITE 9,(127,48),15,8
870 CS=9
880 SWAP U,W:Z=-Z:C=-C
890 NEXT G
900 PUT SPRITE 9,(127,54),15,8
910 GOTO 240
920 A$(0)="d10bd6d4bd46d16"
930 A$(1)="bd2d8bd6d4bd42d24"
940 A$(2)="bd4d4bd8d4bd38d8bd16d
8"
950 A$(3)="bd14d6bd34d8bd24d8"
960 A$(4)="bd12d10bd28d8bd32d8"
970 A$(5)="bd8d16bd22d8bd40d8"
980 A$(6)="bd2d12bd6d6bd2d2bd2d2
bd2d2bd4d8bd48d8"
990 A$(7)="d46bd56d4"
1000 A=0:B=14:C=2:Z=0
1010 FOR T=0 TO 1
1020 FOR X=A TO B STEP C
1030 FOR H=0 TO 1
1040 Z=Z+1
1050 PSET (80+Z+T*32,54),4
1060 DRAW"c1s4"
1070 DRAW A$(X/2)
1080 NEXT H
1090 NEXT X
1100 A=14:B=0:C=-C
1110 NEXT T
1120 DRAW"bm79,64u36r67d36l1u351
65d35"
1130 DRAW"bm97,160r32u1132"
1140 CIRCLE (112,128),25,1,,,1.4
1150 DATA 3E,1C,8,8,8,8,8,8
1160 DATA FF,FF,FF,FF,FF,FF,FF,F
F
1170 SPRITE$(23)=S$:S$=""
1180 DATA 0,0,0,0,0,0,0,0
1190 DATA FF,FF,FF,FF,FF,FF,FF,F
```



```
F
1200 DATA FF,FF,FF,FF,FF,FF,FF,F
F
1210 DATA FF,FF,FF,0,0,0,0,0
1220 DATA FF,FF,FF,FF,FF,FF,FF,F
F
1230 DATA FF,FF,FF,FF,FF,FF,FF,F
F
1240 DATA 3F,3F,FF,FF,FF,FF,FF,F
F
1250 DATA FF,FF,FF,FF,FF,FF,FF,F
F
1260 DATA FF,FF,FF,FF,FF,FF,FF,F
F
1270 DATA FF,FF,FF,0,0,0,0,0
1280 DATA FF,FF,FF,FF,FF,FF,FF,F
F
1290 DATA 0,0,0,0,0,0,0,0
1300 DATA FF,FF,FF,FF,FF,FF,FF,F
F
1310 DATA 0,0,0,0,0,0,0,0
1320 DATA 0,0,0,2C,4A,51,AA,49
1330 FOR I=1 TO 8: READ Q$: S$=S
$+CHR$(VAL("&H"+Q$)):NEXT
1340 SPRITE$(8)=S$:S$=""
1350 FOR I=1 TO 24: READ Q$: S$=
S$+CHR$(VAL("&H"+Q$)):NEXT
1360 SPRITE$(9)=S$:S$=""
1370 FOR J=10 TO 12
1380 FOR I=1 TO 32: READ Q$: S$=
S$+CHR$(VAL("&H"+Q$)):NEXTI
1390 SPRITE$(J)=S$:S$=""
1400 NEXT J
1410 FOR I=1 TO 8: READ Q$: S$=S
$+CHR$(VAL("&H"+Q$)):NEXTI
1420 SPRITE$(23)=S$:S$=""
1430 GOTO 170
1440 IF STICK (0)=1 THEN C=C+SGN
(C)
1450 IF STICK (0)=5 THEN C=C-SGN
(C)
1460 IF C>10 OR C<-10 THEN C=10*
SGN(C)
1470 IF C=0 THEN C=SGN(Z)
1480 IF STRIG (0)=-1 THEN 1480
1490 RETURN
```

GRAFICADOR DE ALTA RESOLUCION

Seguimos afirmando que para entrar en el campo del software no hay mejor forma que comenzar a dar los primeros pasos programando. Es más divertido y se le pierde el miedo a esta tarea, que no es tan complicada como su fama lo adjudica.



Vamos a explicar paso a paso el procedimiento que debemos realizar si queremos lograr un programa. Hemos pensado, un utilitario que nos ayude a dibujar. Así podremos crear dibujos en otros programas, pues puede guardar las posiciones de los puntos estratégicos del diseño. Por ejemplo, si nuestro gráfico consiste en un cuadrado, los puntos claves serán los cuatro vértices. Guardando las posiciones de estos últimos, sólo tendremos que unirlos para obtener nuevamente el mismo cuadrado.

No pretendamos que sea un excelente programa desde el principio, pero sí que cumpla los requisitos básicos del propósito planteado. Los puntos principales del programa son:

1— debemos movernos por la pantalla a través de algún mando como joystick o teclado. Pero como no todos los usuarios cuentan con los primeros, nos moveremos con los cursores del teclado.
2— poder dibujar (marcar o pintar) a medida que nos desplazamos por la pantalla con las teclas, simulando estar dibujando con un lápiz que se mueve sobre un papel.
3— poder borrar.
4— guardar nuestro dibujo.
5— reproducir nuestro diseño.

Según estos puntos, cada vez que nos movamos por la pantalla dejaremos marcados nuestros pasos. En decir que estamos siempre con el lápiz sobre el papel y pintando cada movimiento que éste haga. Obviamente no es lo adecuado. Agreguemos otro punto que resuelva este problema.

6— agregar una opción que al presionar una tecla no pinte nuestro recorrido o movimientos. O sea, levantar el lápiz.

El punto 3 se refiere al borrado del dibujo, pero podríamos hacernos dos preguntas:

¿Qué sucedería si quisiéramos borrar un sector del dibujo? No sería conveniente borrar todo el diseño porque perderíamos partes del mismo que tal vez nos interesen.

¿Cómo haríamos si apenas quisiéramos borrar un par de puntos solamente?

De acuerdo con estos dos planteos, una manera de solucionarlos sería abrir el punto 3 en dos ramas:

3a— borrar un sector (porción medianamente grande)
b— borrar punto por punto. (borrando los puntos que tocamos en nuestro recorrido)

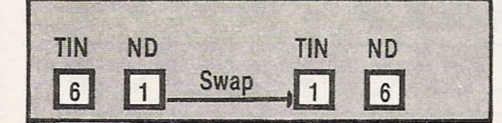
Por ahora tenemos un programa planteado con siete items que deberemos desarrollar.

Pero no significa que no se le puedan ir agregando otras opciones si lo creemos conveniente.

Ahora hay que pensar una forma para que mientras estamos dibujando, podamos escoger cualquiera de las ventajas que mencionamos arriba.

Pensamos que la manera más comoda y práctica sería que al presionar una tecla ya tengamos una opción. O sea, que presionando distintas teclas se vayan ejecutando las distintas opciones. Para esto, a cada una de las teclas de Función les daremos una de las tareas que pro-

FIGURA 1



pusimos. El punto I solamente lo podrán realizar las teclas de cursos. Es decir que si presionamos la tecla que mueve hacia arriba, nos hará correr una posición por encima del lugar donde estábamos.

Tenemos varias tareas para programar, y cada una es independiente de las otras. por lo tanto iremos desarrollando una a la vez.

Podemos comenzar por cualquiera de las siete tareas, empecemos por ejemplo por 3 b.

BORRAR PUNTO POR PUNTO

Si queremos pintar, es lo mismo que decir que iremos fijando pixels con algún color. Al comenzar cada programa, debemos fijar el color de fondo de la pantalla y el que vamos a utilizar para visualizar letras y gráficos, llamado color de tinta.

Esto se logra con la sentencia "COLOR color de tinta, color de fondo". Cada color tiene un código que lo representa. Si queremos color negro de fondo, busquemos —en el manual— cuál es el código destinado para éste. Según nuestro manual, el código I significa color negro y si quisiéramos color de tinta rojo oscuro, el código es 6. Entonces la sentencia sería: COLOR 6,1. Pero pensemos que para obtener el efecto de borrado punto por punto, sería lo mismo que ir pintando cada punto con el mismo color del fondo. De esta manera, los puntos no deseados desaparecerán a medida que pasemos por encima de ellos.

Si estamos marcando puntos con color rojo sobre un fondo negro, borrar dichos puntos sería pintarlos del mismo color del fondo, o sea: negro.

Esto significaría solamente cambiar el código de tintá de al sentencia COLOR para que coincida con el de fondo.

Hay varias formas de realizarlo. Por ejemplo, si guardamos en una variable el color de tinta, cambiaremos el valor de ésta igualándola con el color de fondo.

La instrucción "COLOR TIN" hará que al ir variando de valor TIN, cambie el color de tinta.

Necesitamos dos variables: TIN y ND que guarden el código del color negro y del rojo oscuro. Siempre se va a pintar con el color que represente TIN, mientras que el color de fondo se mantiene negro constantemente.

Si queremos que los puntos que vamos trazando sean rojos (para que resalten sobre el fondo negro), la variable TIN deberá tener almacenado el número 6. Pero si queremos borrar los puntos, TIN debe tener almacenado el valor I para coincidir con el fondo y simular desaparecer.

Al comenzar el programa, TIN tendrá el valor 6 (para poder pintar puntos visibles, que resalten del fondo) y ND el valor I.

Si al presionar una tecla se intercambian los valores de TIN y ND, es decir que en TIN queda almacenado el color de fondo (I) y en ND: el de tinta (6). Al ejecutar la sentencia "COLOR TIN" pintará los puntos con el mismo color de fondo, simulando así el borrado de los puntos por donde pasamos.

La instrucción "SWAP variable I, variable 2" hará que el valor de la variable I se almacene en la variable 2, mientras que el valor de la 2 queda guardado en la I.

SWAP TIN, ND hará que el valor que tenía en TIN sea almacenado en ND, al mismo tiempo que el valor que tenía ND paso a TIN. Ver figura I.

Entonces cada vez que presionemos una tecla de función (por ejemplo F1), intercambiaremos el color de tinta.

Para programar entonces la rutina de borrar punto por punto, escribiremos una línea con dos sentencias. Una que intercambie los colores y la otra que fije en la instrucción COLOR, el tono que se va a utilizar.

Cómo esta opción de borrar será llamada como un subrutina, esto significa que el programa saltará a una línea especificada, realizará las operaciones que encuentre hasta una sentencia RETURN y luego volverá a la siguiente orden desde donde se produjo el salto o llamada a la subrutina. En cada final de la subrutina es necesario colocar una sentencia RETURN justamente para indicar que ése es el final.

Toda la subrutina que permite borrar punto por punto, quedaría de la siguiente manera:

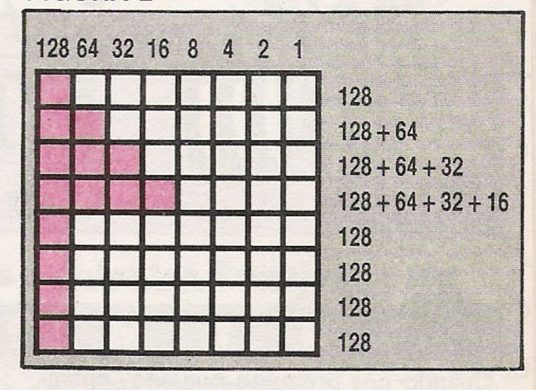
(I) SWAP TIN, ND: COLOR TIN: RETURN

Con esto ya conseguimos programar una porción del programa que más adelante veremos como unirlo con el resto.

FABRICAR UN SPRITE

Para ir moviéndonos por la pantalla, necesitamos una guía. Algo que nos vaya marcando nuestra posición en la pantalla, como una flecha, punto o cualquier otro dibujo.

FIGURA 2



Vamos a definir para esto un SPRITE. Esta parece ser una palabra mágica. Muchos pueden pensar que vamos a realizar algo complicado. Esto se debe tal vez a que en otras computadoras, definir Sprites es complicado y en consecuencia, su fama de ser una tarea destinada solamente para expertos en programación.

Pero desde nuestro primer número, hemos aclarado que una de las ventajas de las MSX es facilitar al usuario la definición de Sprites. Para refrescar cómo se definen este tipo de gráficos, echemos un vistazo al primer número de MSX (pág. 20) o bien leamos del manual de la máquina cómo se utilizan las funciones para Sprites.

Si queremos que nuestro dibujo de guía sea una flecha como el de la figura 2, calculemos cuanto vale cada fila de bits. La primera valdrá 128, la segunda 192, la tercera 224, la cuarta 240 y las cuatro últimas 128.

Luego debemos cargar cada uno de estos valores en una variable caracter de la siguiente forma:

realizar 8 veces ___ FOR F = I TO 8
leer un dato ____ READ A
y guardarlo en la variable A

guardar el valor de A convertido en caracter ________ S$ = S$ + CHR$(A)
definir el sprite "0" ______ SPRITE$(0) = S$
con el gráfico definido

colocar los datos para ser leídos __________ DATA 128, 192, 224, 240, 128, 128, 128, 128

Este también es un sector del programa, pero debe ir al comienzo del mismo porque lo necesitamos para empezar a graficar.

Hemos visto una parte del utilitario. Debido a que su explicación es demasiado larga para una sola nota, continuaremos con la misma en el siguiente número.



CALENDARIO ETERNO

Clase: Utilitario



Este programa nos mostrará por pantalla los meses del año que querramos.

Cuando nos pregunte por año, entremos sólo los dos últimos dígitos. Si nos queremos referir al año 1986, debemos entrar el número 86 solamente.

A continuación preguntará por el mes que deseamos ver. Contestemos con el número del mes a observar. Si entramos cero (0), aparecerán todos los meses comenzando por enero.

En el margen izquierdo aparecerá la cantidad de días que pasaron en el año, tomando como referencia el primero de enero hasta el primero del mes que estamos viendo. Mientras que la cifra del margen derecho corresponde a los días comprendidos entre el primero del mes en observación hasta el último del año.

Si queremos sacar los meses por impresora, agreguemos a todas las sentencias "PRINT", el prefijo "L" quedando entonces "LPRINT".

```
           JULIO' 86

181                              184

    L  M  M  J  V  S  D

       1  2  3  4  5  6
    7  8  9 10 11 12 13
   14 15 16 17 18 19 20
   21 22 23 24 25 26 27
   28 29 30 31
```

VARIABLES IMPORTANTES

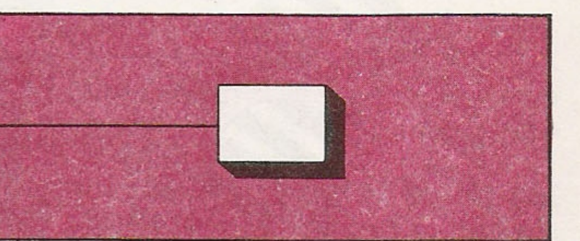
M$: matriz con los meses
N$: matriz con la cantidad de días de cada mes.
U: año que deseamos ver
V$: mes que queremos ver
B: si B = 1 el año es bisiesto, si B = 0 no lo es

ESTRUCTURA DEL PROGRAMA

30-90: inicializa matrices
110-150: pregunta sobre las fechas que deseamos ver
170-410: rutina para imprimir un mes en particular
430-610: imprime todos los meses
630-680: calcula si el año es bisiesto y qué día cae el primero del mes

```
10 ' CALENDARIO
20 '******************************
****
30 DIM M$(12),N(12)
40 DATA ENERO,31,FEBRERO,28,MARZ
O,31
50 DATA ABRIL,30,MAYO,31,JUNIO,3
0,JULIO,31,AGOSTO,31
60 DATA SEPTIEMBRE,30,OCTUBRE,31
,NOVIEMBRE,30,DICIEMBRE,31
70 FOR I=1 TO 12
80 READ M$(I),N(I)
90 NEXT I
100 ' *************************
******
110 SCREEN 0:COLOR 6,1,1: KEY OF
F
120 INPUT "AÑO";U
130 IF U>99 THEN 120
140 LOCATE 0,9:INPUT"QUE MES QUI
ERE VER: (0 PARA VER TODO EL AÑO)
 ";V$
150 IF V$="0" THEN GOTO 430
160 ' *************************
******
170 GOSUB 630:CLS:PRINT TAB(11);
M$(VAL(V$))+"'";U:PRINT:PRINT
180 FOR I=1 TO VAL(V$)-1
190 NJ=N(I)
200 IF I=2 AND B=1 THEN NJ=29
210 FOR K=1 TO NJ
220 K$=RIGHT$(STR$(K),2)
230 Y=Y+1
240 IF Y=8 THEN Y=1:
250 NEXT K
260 J=J+NJ
270 NEXT I
280 NJ=N(VAL(V$))
290 IF V$="2" AND B=1 THEN NJ=29
300 PRINT TAB(2);J;TAB(30);365+B
-J
310 PRINT: PRINT
320 PRINT TAB(9);"L  M  M  J  V
 S  D"
330 PRINT :PRINT:PRINT
340 FOR K=1 TO NJ
350 K$=RIGHT$(STR$(K),2)
360 PRINT TAB((Y-1)*3+8);K$;
370 Y=Y+1
380 IF Y=8 THEN Y=1: PRINT:PRINT
390 NEXT K
400 J=J+NJ
410 A$=INPUT$(1):GOTO 110
420 ' *************************
*****
430 GOSUB 630:FOR I=1 TO 12
440 NJ=N(I)
450 IF I=2 AND B=1 THEN NJ=29
460 CLS: PRINT
470 PRINT TAB(11);M$(I)+"'";U:PR
INT:PRINT
480 PRINT TAB(2);J;TAB(30);365+B
-J
490 PRINT: PRINT
500 PRINT TAB(9);"L  M  M  J  V
 S  D"
510 PRINT :PRINT:PRINT
520 FOR K=1 TO NJ
530 K$=RIGHT$(STR$(K),2)
540 PRINT TAB((Y-1)*3+8);K$;
550 Y=Y+1
560 IF Y=8 THEN Y=1: PRINT:PRINT
570 NEXT K
580 J=J+NJ
590 A$=INPUT$(1)
600 NEXT I
610 GOTO 110
620 ' *************************
****
630 J=0: B=0
640 M=(U*365)+INT((U-1)/4)
650 IF U=0 THEN M=0
660 Y=M-(7*INT(M/7))+1
670 IF INT(U/4)=U/4 THEN B=1
680 RETURN
```

CONCURSO DE PROGRAMAS



auspiciado por TELEMATICA S.A. que proveerá los siguientes Premios:

P R I M E R   P R E M I O

Un periférico (a elección entre un monitor, una diskettera y una impresora).

Una Beca para trabajar en el Departamento de Investigación y Desarrollo de Telemática S.A.

S E G U N D O   P R E M I O

Un periférico (a elección entre un monitor, una diskettera y una impresora).

En caso de que el ganador no pueda utilizar la beca será ofrecida a quien obtenga el segundo premio, y si este tampoco pudiera aprovecharla se otorgará a alguno de los participantes del certamen que se hubiera destacado.

ESPECIAL Entre los programas recibidos, algunos de ellos podrán ser editados por Prosoft, reconociéndose los derechos de autor.

J U E G O S :

Temas: ● **TRUCO.** Premiaremos al programador que logre la mejor versión de este tradicional juego de salón.

● **DE INTELIGENCIA.** Los juegos que nos inspiren podrán ser "El Estanciero" o temas originales que sigan la línea

E D U C A T I V O S :

Tema: ● **LIBRE.**

U T I L I T A R I O :

Tema: ● **INTELIGENCIA ARTIFICIAL.** Se trata de lograr un programa que realmente nos haga discutir sobre el límite entre la simulación y la inteligencia de la computadora.

Obviamente la única forma de lograr esto será siguiendo las reglas propuestas de la Inteligencia artificial. Por esto se considerarán para premiar esta categoría, además de las reglas detalladas más abajo, la capacidad de auto-aprendizaje del programa, el nivel de inferencia del mismo, la capacidad y modo de almacenamiento de su base de datos, y principalmente su analizador sintáctico dado que hasta el momento no se ha logrado uno que dé suficiente credibilidad de que estamos frente a un ser racional que entiende nuestro idioma y no frente a una máquina a la que debemos hablarle con verbos en infinitivo al mejor estilo Tarzán (sin menospreciar a este último).

PROFESIONAL O GESTION:

Tema: ● **LIBRE.** Dentro de este tema podrán figurar programas de las más diversas aplicaciones.

BASES: No sólo será indispensable que el programa enviado funcione correctamente sino que además debe cumplir con ciertas reglas.

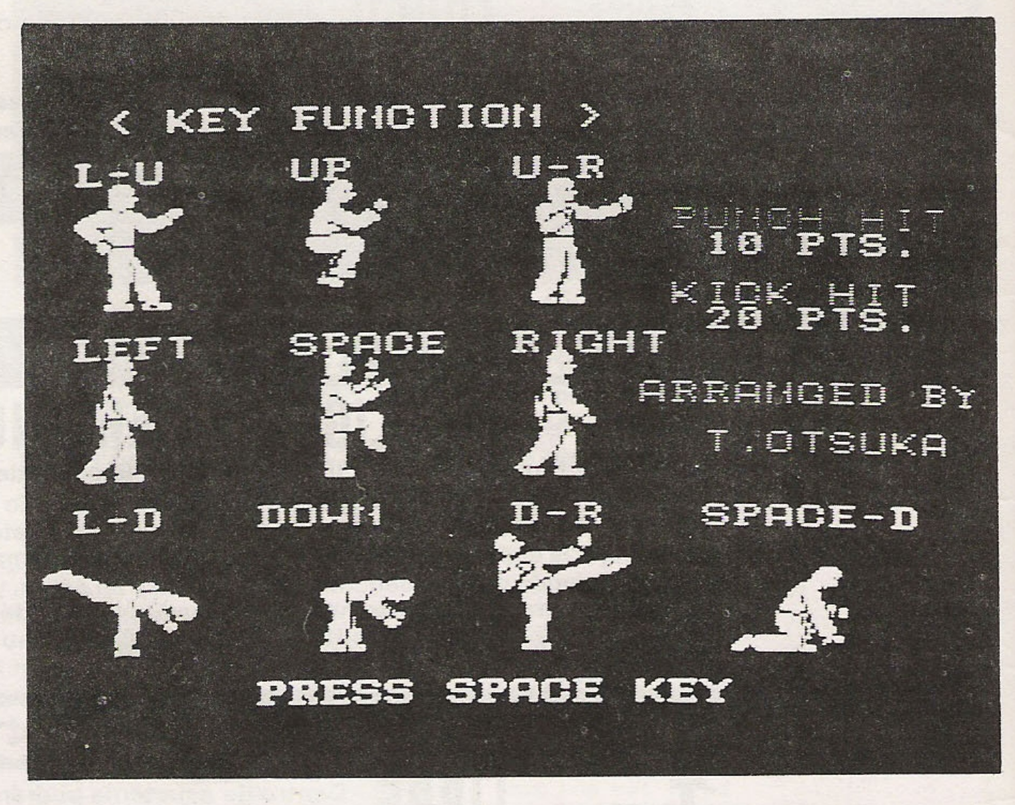
- *Programación estructurada en bloques fácilmente diferenciables.*
- *Fácil seguimiento del mismo y detalle de éste como parte de su documentación. (Diagrama de bloques con los números de línea que los identifiquen).*
- *Aclaración y clara explicación de los algoritmos utilizados, deben figurar como parte de la documentación.*
- *Las variables y/o direcciones de memoria utilizados también se deben incluír en esta documentación.*
- *Listado de nemónicos assembler y la localización en memoria si es que se utiliza este tipo de lenguaje.*
- *Calidad y originalidad de gráficos, sonidos y pantallas de menú.*
- *El programa debe ajustarse a alguno de los temas propuestos más arriba. Esto es ELIMINATORIO.*

Raiting Soft

PROYECT A

CREATIVIDAD: 7
PRESENTACION: 8
ATRACCION: 9
SONIDO: 7
TIPO: ENTRET
PRODUCE: BITGAME





Con este entretenido juego nos convertiremos en Jakie, luchador de artes marciales, y deberemos salvar a nuestra amada secuestrada por los Ninjas y alojada en el templo de éstos.

No tenemos armas para defendernos salvo nuestra propia habilidad en este tipo de luchas orientales.

Al comenzar el entretenimiento, desembarcamos en el templo de los secuestradores. Inmediatamente nos atacarán vampiros que vuelan a distintos niveles. Podemos saltarlos o matarlos con golpes de pies (si vuelan alto) y de puño (si vuelan a la altura del hombro). En cambio los podremos estrangular si vuelan a la altura de nuestros pies.

Contamos con una cantidad limitada de energía, y la perdemos cuando nos detenemos o somos mordidos por estos mamíferos.

Pero si observamos que empezamos a debilitarnos, caminemos hacia adelante o atrás para recuperarnos.

Una vez que hayamos pasado este enfrentamiento con los vampiros, los luchadores Ninjas tratarán de destruirnos. Cada golpe de ellos que nos toque, consumirá bastante energía y al acabarse ésta, perderemos una vida.

En la primer etapa de la lucha contra estos guerreros, deberemos derribar a cuatro discípulos y luego al maestro, siendo este último más difícil de derrotar.

Los golpes pueden ser varios. Nosotros no conocemos los nombres de los movimientos, pero la patada arriba y adelante se consigue moviendo abajo y a la derecha al mismo tiempo; el golpe de puño hacia adelante se logra con un movimiento hacia arriba y a la derecha simultáneamente. Pero para patear atrás y arriba, movamos a la izquierda y abajo.

Hay otros movimientos que se consiguen combinando las cuatro direcciones básicas (arriba, abajo, derecha e izquierda), más el botón de disparo (si jugamos con joystick) o la barra de espacio (si lo hacemos con los cursores del teclado).

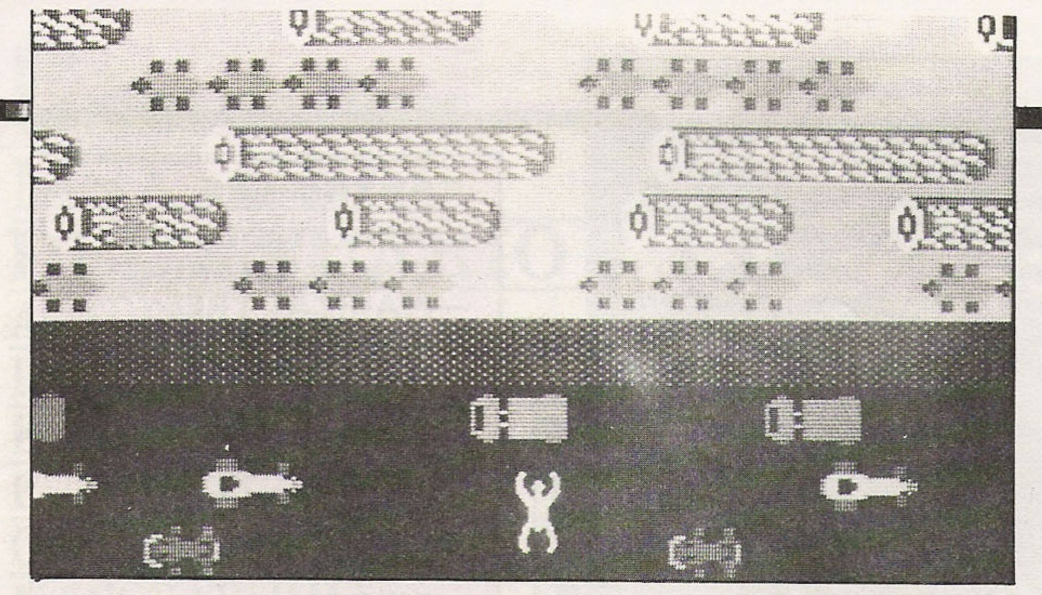
Al golpear varias veces a un luchador, lo derribaremos sin matarlo, pero cuando está en el suelo es nuestra oportunidad de destruirlo ahorcándolo. Para esto hay que presionar la barra de espacio y el cursor que mueve hacia abajo, si jugamos con el teclado, o bien presionar el botón de disparo y mover la palanca para abajo si jugamos con joystick.

Si al estar levantándose el guerrero le damos un golpe con el pie, también lo derribaremos definitivamente.

Cada vez que derrotemos a un maestro, pasamos a otro nivel con discípulos más expertos, mayor cantidad de vampiros y en consecuencia más dificultades.

Lo importante es estar atentos a la energía que tengamos, tratando de caminar en cualquier dirección para reabastecernos.

En el programa están las instrucciones pero no creo que muchos las entiendan pues están escritas en un idioma oriental. No será demasiado difícil interpretar cómo hacer los movimientos, porque tienen la explicación de cómo se logran. Estas pantallas están al comienzo del programa, y se ven si no presionamos ninguna tecla. Solamente esperemos un ratito y aparecerán.



FROGGER

CREATIVIDAD: 7
PRESENTACION: 7,
ATRACCION: 8
GRAFICOS: 8
SONIDOS: 8
*TIPO: **ENTRETENIMIENTO***
*PRODUCE: **BITGAME***



Este divertido juego se trata de una ranita muy simpática que insiste en querer llegar al otro extremo de la calle, subir a troncos y hojas que flotan en el río y así llegar a su cueva.

Pero las hojas suelen hundirse; por lo tanto la ranita no deberá apoyarse en ellas porque si no se ahogará.

Los troncos son más seguros porque se mantienen siempre flotando.

Cruzar la transitada avenida no es sencillo, los autos y camiones pasan a diferentes velocidades y por cuatro carriles.

Contamos para llegar a la cueva con un tiempo límite. Al agotarse éste, perderemos una vida.

Este juego permite usar joysticks o los cursores del teclado, y pueden jugar uno o dos jugadores.

Para completar cada nivel, tenemos que llevar a la ranita hasta cada cueva. Cada nivel presenta mayor dificultad como: más rapidez de los autos, menor cantidad de hojas y troncos, desincronización entre ambos.

Contamos solamente con tres vidas para realizar la mayor cantidad de puntos posible.

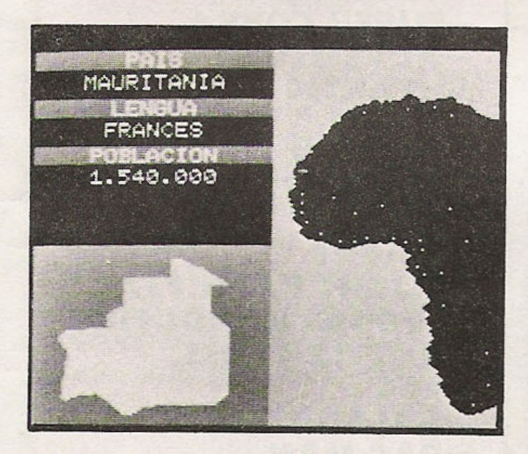
Los gráficos son muy buenos, la ranita está muy lograda al igual que los autos. Si perdemos una vida, aparecerá un símbolo de muerte por cierto cómico como lo es también la cara de satisfacción de la rana cuando llega a su cueva.

En el segundo nivel, si subimos en el mismo tronco que otra rana y conseguirmos llegar hasta alguna casilla, obtendremos el doble de puntaje.

La alegre musiquita que acompaña al entretenimiento sólo la podremos escuchar al comienzo y final del juego.

Es destacable que FROGGER se trata de uno de los pocos pasatiempos donde no hace falta matar a nadie para poder ganar.

AFRICA





CREATIVIDAD: 7
PRESENTACION: 7,
ATRACCION: 8
GRAFICOS: 8
SONIDOS:
*TIPO: **ENTRETENIMIENTO***
*PRODUCE: **BITGAME***



Si estudiar el continente africanó en su parte política nos re-

programa hará que nuestro aprendizaje sea más ameno.

Este soft permitirá conocer las características geopolíticas del continente en forma de juego, describiendo los puntos más importantes en su estudio como: país, capital, otra ciudad importante, moneda utilizada, lengua y población.

Cada nación es representada gráficamente, acompañando la información de la misma.

las escuelas primarias, porque para este tipo de estudiantes, debe existir un estímulo que provoque cierto interés y este programa creemos que lo cumple.

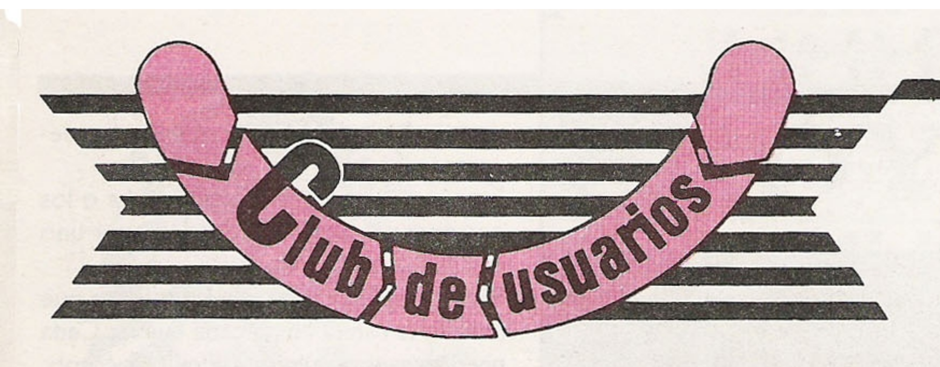
Al comenzar se verá el mapa de Africa, pero debemos esperar que desaparezca la indicación que el programa se sigue cargando.

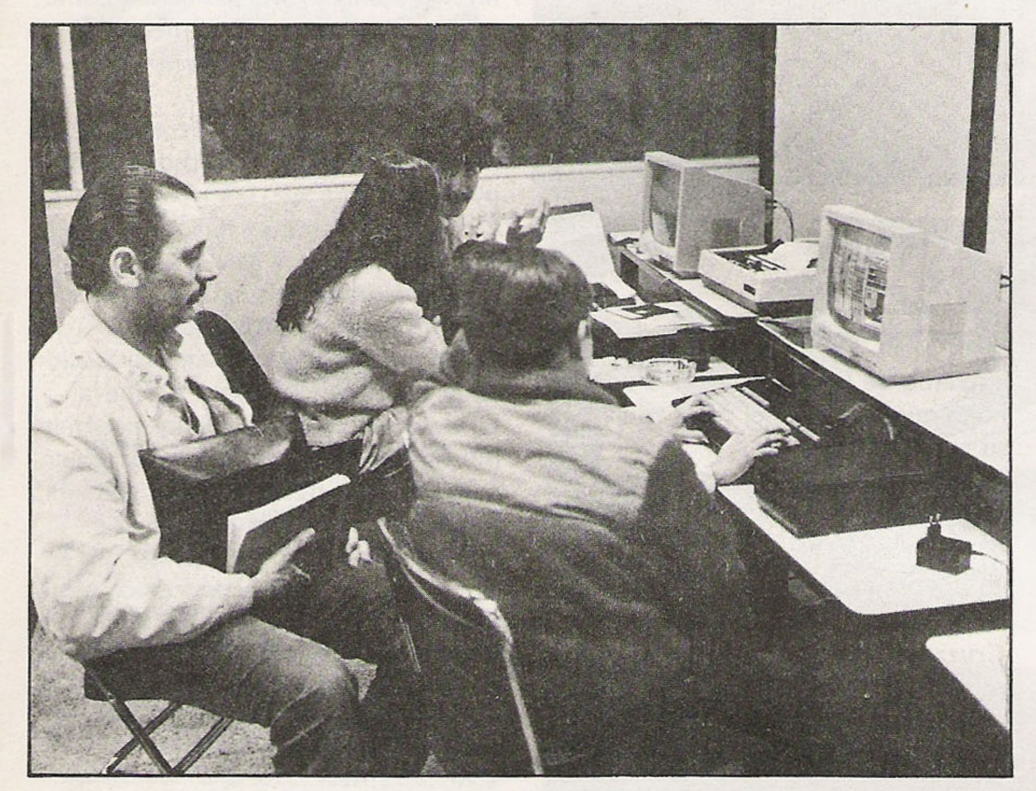
Una vez completada esta operación, se ubica una cruz en el país del que se requiere información, a través de los cursores.

Al presionar la barra de espacio, se escribirá el nombre de la nación donde estamos posicionados. En cambio si presionamos RETURN, aparecerá el gráfico del país con los datos elementales como: capital, otra ciudad importante y moneda.

Presionar la barra de espacio una vez que esta información se imprimió, hará que se completen los datos, describiendo el idioma y la cantidad de población. Pero si hubiéramos presionado RETURN en vez de la barra de espacio, volveríamos al mapa para elegir otra ciudad.

Este programa no tiene sonido, pero es excelente su calidad gráfica. Los contornos de los países y del continente están





ADAPTADOR PARA MSX WRITE

El siguiente listado Assembler permitirá a los poseedores del MSX WRITE que ese programa acepte e imprima en papel, las eñes y los acentos.

Por supuesto, nuestra impresora debe poseer en su set de caracteres estos símbolos.

Así se deben copiar en memoria los bytes de la figura a partir de la posición &HODOOO.

Hecho esto lo probaremos, y lo haremos correr antes de comenzar a editar el texto.

```
33   182   255   54   205   35   54   12   35   54   208
201   254   164   32   3   62   124   201   254   165   32
3   62   92   201   254   160   32   6   205   75   208
62   97   201   254   130   32   6   205   75   208   62
101   201   254   161   32   6   205   75   208   62   105
201   254   162   32   6   205   75   208   62   111   201
254   163   192   205   75   208   62   117   201   62   96
205   165   0   62   8   205   165   0   201   0   0
```

CARTIDGES PARA ALQUILER Y CASSETTES PARA USO INTERNO

En el Club de usarios se encuentra a disposición de los socios, los siguientes juegos para uso interno.

Los cartridges, aparte del uso interno, también se pueden alquilar con las siguientes cifras:

3 días 3A
5 días 4,50A
7 días 6A

CARTRIDGES

- —ATHLETIC LAND
- —CIRCUS CHARLIE
- —MAGICAL TREE
- —COMIC BAKERY
- —HYPER RALLY
- —TENNIS
- —SKY JAGUAR
- —KING'S VALLEY
- —ROAD FIGHTER
- —PING PONG
- —YIE-AR-KUNG FU
- —PIPPOL
- —SOCCER
- —BOXING
- —ANTARTIC ADVENTURE
- —GOLF
- —HYPER SPORTS 1
- —HYPER SPORTS 2
- —MONKEY ACADEMY
- —SUPER COBRA
- —TRACK & FLIELD 1
- —TRACK & FLIELD 2
- —TIME PILOT

CASSETTES

- —MANIC MINER
- —SORCERY
- —JEGUAR
- —BOMBERMAN
- —MONKEY ACADEMY
- —H.E.R.O.
- —TIME PILOT
- —SUPER COBRA
- —ADVENTURE ANTARTIC
- —TOEMI
- —AJEDREZ
- —EL EJECUTIVO
- —GOLF
- —MAPPY
- —FLIGHT PATH 737
- —SPOOKS & LADDERS
- —BATTLE CROSS
- —EGG
- —GALAGA
- —PAC MAN
- —FROGGER
- —EXERION
- —WARD HEAD
- —PROJECT A
- —SPARTAN X
- —CAPTAIN CHEF
- —WEDDING BELLS
- —BANG BANG
- —DRILLER TANK
- —STAR ADVENGER
- —VACUMANIA
- —RIVER CHASE
- —SENJYO

MSX. GUIA DEL USUARIO

AUTOR: PAUL HOFFMAN

EDITORIAL OSBORNE MACGRAW-HILL





Este libro contiene una introducción sencilla al mundo de la utilización de estas máquinas estandar.

Pero no es solamente útil para los que recién se inician en el manejo de las MSX, sino también para los expertos programadores de este sistema porque explica algunos conceptos y trucos que no vienen en los manuales de las computadoras.

Es el complemento ideal de estos manuales porque ayudan a entender las explicaciones confusas que suelen acompañar a las máquinas.

En este libro podemos encontrar respuestas sobre qué se puede hacer con las MSX, aparte de juegos, cómo familiarizarnos y manejar este tipo de BASIC.

También para los usuarios de unidades de disco, hay una explicación profunda sobre todo lo referido a este tema como qué es un disco, qué hace y cómo se utiliza el MSX-DOS.

Tiene una sección de información rápida con las explicaciones necesarias para poder usar correctamente las sentencias del MSX-BASIC y las del MSX-DOS.

Está incorporada a este completo libro, la tabla de los códigos ASCII, que se encuentran ausentes en la mayoría de los manuales de las computadoras.

Para los programadores de Assembler, hay un apéndice con las posiciones útiles de la memoria RAM. Aquí se indica en qué lugar de la memoria está almacenado el código de color para el fondo y tinta. De esta forma podremos cambiar con POKE o desde el lenguaje de máquina, los colores.

Y para los más adelantados en cuanto a conocimientos técnicos, tienen a su disposición toda la información necesaria sobre el Bus de Cartuchos.

Este libro nos dará pautas para ir avanzando en el mundo de la computación, porque nos explica desde cómo realizar nuestro primer programa, como entender o seguir un listado, hasta crear algoritmos en el lenguaje ensamblador.

Tampoco deja sin comentar lo referido a utilización de gráficos, sonido, joysticks y manejo de entrada y salida de datos.

Gran parte de la información que se recolectó para escribir este libro fue suministrada por gente de MICROSOFT CORPORATION y de otras personalidades del ambiente.

GUIA DEL PROGRAMADOR MSX

AUTOR: C.I. BURKINSHAW y R. GOODLEY

EDITORIAL: RA—MA



Está dirigido a los usuarios de las MSX con conocimientos avanzados sobre este tipo de máquinas.

Es un completo compendio donde encontraremos las explicaciones del funcionamiento de nuestro computador. Tiene temas interesantes como el lenguaje de máquina para el Z-80, una explicación detallada del procesador de video y de sonido, y también todo sobre la entrada y salida de joysticks, teclado y pantalla.

Tampoco escapa a este libro una minuciosa exploración sobre las funciones del BASIC MSX, como el diseño de sprites y gráficos en alta resolución.

Trae apéndices de información rápida sobre: código de caracteres, código de colores, tabla de VRAM, instrucciones del Z-80, el chip de pantalla TI- 9929 y del chip de sonido GI AY-3-8910.

Para comprender su contenido es aconsejable contar con los conocimientos básicos sobre este tipo de computadoras, y es ideal para aquellos que les interese la parte técnica de las MSX.



Para comunicarse con nosotros deben escribirnos a Revista para usuarios de MSX, Paraná 720, 5to. Piso, (1017) Cap. Federal.

PADDLE

Los felicito por la decisión de haber sacado una revista para los usuarios de MSX, es como un caldo de sopa concentrado y sin desperdicios (espero que sigan en esa línea), no me gustan las revistas que son puro programas. Les digo como usuario qué me interesa: código de máquina, trucos para aprovechar a fondo la máquina como eso del F% de la pág. 21 del primer número, circuitos eléctricos de aplicación, como memorias EPROM y cómo instalar un reset.
Tengo preguntas: ¿qué es un paddle, por qué se les dice "no enmascarables" a ciertas interrupciones y por qué las grabaciones en ASCII son más lentas?
Me gustaría además (anticipándome) que le dieran mucha importancia al correo de lectores; y que las explicaciones de ustedes sean mejores que las del manual técnico en relación a las funciones play y sound.
Escuché nombrar el juego del "estanciero", pero no lo conozco, ¿podrían explicarme de qué se trata?

Claudio Herrero - E. Zeballos 3589 - V. Domínico - Buenos Aires.

LOAD MSX

Sobre tu primer interrogante, te contamos que el paddle es una interfase muy utilizada como joystick, pero no se trata de una palanca que se mueve, sino de una especie de cilindro que se hace girar. Es empleado generalmente para los programas con coches de carrera porque el paddle simula ser el volante.
Interrupciones "no enmascarables" se les llama a aquellas que deben ser tratadas en el mismo momento de la interrupción porque no permiten posponer su procesamiento.
Y en cuanto a tu tercer pregunta, es cierto que las grabaciones en ASCII son lerdas debido a dos motivos principales:
1) todas las instrucciones tienen un número que las representan. Si no grabamos en ASCII, la computadora guarda un solo número por cada instrucción. Por ejemplo, supongamos que se está grabando la instrucción PRINT. Si lo hacemos en ASCII, guarda el valor que corresponde a la P, luego a la R, después a la I, sigue con el código de la N y por último la T. En cambio si lo grabamos de la otra manera, solamente graba un número que representa toda la instrucción PRINT y esto obviamente ahorra tiempo.
2) al grabar en ASCII, se van guardando sectores de 256 bytes, cada uno separado por un pulso, ocupando más espacio.
Como esta es una revista dedicada a sus lectores, por supuesto que la sección MAILING tiene mucha importancia y nuestra idea es que así se mantenga.
Te contamos brevemente que el "estanciero" se trata de un juego para varios participantes donde cada uno comienza teniendo solamente un capital en efectivo. El tablero del juego está dividido por provincias, que se pueden comprar si caemos (por medio de los dados) en ese territorio. Cuando algún contrincante cae en nuestras tierras deberá pagarnos el alquiler. Entonces la finalidad del juego es incrementar nuestro capital y tratar que nuestros compañeros de juego queden en quiebra.

Claudio también nos envió su teléfono: 203-2903 porque le gustaría intercambiar datos e información con otros lectores.

Y muchas gracias por las sugerencias de los temas que se podrían llegar a tratar, lo estamos haciendo y esperamos ampliar los temas que propusiste entre otros más.

BLOQUEO DE LA DISKETERA

Les escribo estas líneas a fin de felicitarlos por su nueva y muy necesaria revista para los que, como en mi caso tenemos una de estas "maquinitas".
También les escribo para preguntarles por qué cuando se carga un programa en lenguaje de máquina, la luz de "in use" de la disketera permanece encendida hasta que se apaga la máquina (en mi caso una SPECTRVIDEO).

Ricardo G. Pesce - Constitución 984 2° "F" - San Bernardo - Buenos Aires

LOAD MSX

Tu problema es muy común y sucede en casi todas las marcas. Resulta que la mayoría de los programas en código de máquina utilizan desde el Assembler la disketera y la dejan habilitada para grabar o cargar. Entonces el motor continúa funcionando, pero sin escribir ni leer ninguna información del diskette.
En algunos programas este funcionamiento del motor se produce involuntariamente, pero no nos preocupa demasiado porque el disco no se afectará.

LIBRO ASSEMBLER

Quiero felicitarlos por la revista, que es muy interesante, ya que nos es muy útil a los usuarios MSX porque nos pone al tanto de todos los avances de hardware y software que produce la empresa Telemática.
Tengo interés en aprender Assembler y quisiera que me aconsejaran algún libro con el cual aprenderlo. También me gustaría que me respondieran lo siguiente: ¿con qué impresora puedo imprimir gráficos (círculos, líneas o pixels) y textos?, ¿qué ventajas y desventajas tiene respecto a las demás impresoras?

Alejandro Barrios - Pedro Seguí 767 - Paraná - Entre Ríos

LOAD MSX

Agradecemos tus felicitaciones porque nos alientan a seguir esmerándonos.
En "Crítica de libros" del número 3 publicamos nuestra opinión sobre uno de los trabajos que abarca el tema que te interesa. También te podríamos recomendar otro: "Lenguaje Máquina para MSX", de Joe Pritchard, de la editorial Anaya.
Contestando tu segunda inquietud, las únicas impresoras capaces de graficar en alta resolución son las denominadas "Plotter".
Hay diversas marcas, pero Talent comercializa una en el país que es compatible con las MSX. Este tipo de impresoras son más costosas que las que imprimen solamente caracteres.

Computación, una oportunidad para que todos enseñen y aprendan.

Un lugar para

desarrollar el pensamiento.

descubrir una vocación.

manejar lenguajes de computación.

comprender los múltiples usos de un computador.

capacitar y perfeccionar al docente.

incorporar los avances tecnológicos.

que el profesional domine el uso de nuevas herramientas.

que los padres se reencuentren con sus hijos.

"No se trata solamente de adquirir en forma puntual conocimientos definitivos, sino prepararse a elaborar a lo largo de toda la vida, un saber en constante evolución y de aprender a ser."

UNESCO

Actividades '86

Para Niños, Adolescentes, Adultos, Docentes, Profesionales y Establecimientos educativos.

- INTRODUCCION A MICROCOMPUTADORES
- DIAGRAMACION ESTRUCTURADA
- LOGO
- BASIC
- COLOR - SPRITE - SONIDO
- COBOL
- PASCAL
- ASSEMBLER
- MS - DOS Y MSX - DOS
- D BASE II - MULTIPLAN
- PROCESADOR DE LA PALABRA
- INSTALACION DE LABORATORIOS

en Establecimientos educativos con formación de multiplicadores y apoyo a la comunidad.

Cómo?

- Taller en grupos de 12 a 15 personas.
- Clases de 2 horas diarias.
- 2 ó 3 alumnos por equipo.
- Equipos disponibles para prácticas adicionales en horarios libres.
- Becas rentadas en el Departamento de investigación y desarrollo de Talent MSX.
- Becas rentadas para docentes en Laboratorios de Establecimientos Educativos.

Informes, Inscripción y Cursos

Lunes a Viernes de 8 a 22 hs.
Sábados de 8 a 13 hs.
CENTRAL:
Cabildo 2027 - 1er. Piso "A" (1428)
Capital Federal
FILIALES:
Centro: Esmeralda 320 - 3º (1343)
Capital Federal
Lanús: Caaguazú 2186
L. Este
Capital: Tucumán 2044 - 1º
(1050) Capital Federal

Talent MSX
Inteligencia en crecimiento.

Cedei
Centro para el desarrollo de la inteligencia.

Descubramos y construyamos juntos los caminos que nos permitirán el uso inteligente de los productos de la creatividad humana.

diálogo - *D.P.*

Titulos no oficiales